DOCUMENT RESUME

ED 294 567 IR 013 234

TITLE	ADCIS Conference Proceedings. International Conference (29th, Oakland, California, November 9-12, 1987).
INSTITUTION	Association for the Development of Computer-based Instructional Systems.
PUB DATE	Nov 87
NOTE	405p.; For the proceedings for the 28th Conference, see ED 278 378.
PUB TYPE	Collected Works - Conference Proceedings (021) -- Reports - Research/Technical (143)
EDRS PRICE	MF01/PC17 Plus Postage.
DESCRIPTORS	Artificial Intelligence; Authoring Aids (Programing); *Computer Assisted Instruction; Computer Graphics; *Computer Managed Instruction; *Educational Technology; Elementary Secondary Education; Health Education; Higher Education; Instructional Design; Interactive Video; *Microcomputers; Special Education
IDENTIFIERS	Software Design

ABSTRACT

These proceedings include papers on such topics as authoring systems, computer-managed instruction, testing, instructional design, management education, simulations, intelligent computer-assisted instruction, and other areas related to computer-based education. Fifty-six papers and 104 abstracts are organized by Association for the Development of Computer-Based Instructional Systems (ADCIS) special interest groups: (1) computer-based training; (2) elementary, secondary, and junior college educators; (3) educators of the handicapped; (4) health education; (5) home economics; (6) implementation and management; (7) interactive video-audio; (8) mini/micro computer users; (9) music educators; (10) PILOT users; (11) PLATO users; and (12) theory and research. (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ADCIS

Association for the
Development of
Computer Based
Instructional Systems

PROCEEDINGS

29th International Conference

November 9 - 12, 1987 Hyatt Regency - Oakland. Oakland, California

ED294567

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality

- Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

IR013234

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

J.J. Hirshbuhl

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Computer - Based Education and Training

29th ADCIS CONFERENCE PROCEEDINGS

Association for the Development of Computer-Based Instructional Systems

OAKLAND, CALIFORNIA
NOVEMBER 9 thru 12, 1987

Copyright ADCIS 1987

ADCIS, MILLER HALL 409
WESTERN WASHINGTON UNIVERSITY
BELLINGHAM, WASHINGTON 98225

PRESIDENT'S MESSAGE

As an interdisciplinary field, educational technology reflects the changes which occur in computer science, instructional design, psychology, and media. One of the most difficult tasks of being a professional in this field is keeping up with this changing technology. As one of the leading professional societies in computer-based education, ADCIS members are involved in virtually every area of CBE activity. These proceedings represent a significant contribution to our field, covering the range of issues, populations, and technologies which are affecting our field today. It is our goal that these proceedings will help you stay abreast of the rapid changes which are occuring throughout the many disciplines of CBE.

These proceedings include papers on such diverse topics as authoring systems, computer managed instruction, testing, instructional design, management education, simulations, intelligent CAI and many more. In addition to these broad areas, the papers in these proceedings represent work in the public schools, universities, business, industry and government, as well as many foreign nations. The topics range from basic "how to" papers and summaries of works in progress, to significant contributions to the theory and research upon which the field is based.

These proceedings are particularly important for two reasons. First, they not only cover the technologies of importance in our field today, they also address, in a single volume, how these technologies can be applied in nearly all areas of education and training. Second, these proceedings represent the most current, up-to-date collection of reports on research and development in our field that is available today. The entire process of solicitation, review, and publication of these proceedings occurs within a span of only nine months, ensuring both a current and relevant set of papers.

The production of these proceedings requires a tremendous effort on the part of the Special Interest Group chairpersons, the Program Chair, and the authors. Equally impressive is the effort put forth by our headquarters staff. Peggy Bishop, and our Executive Director, Gordon Hayes, provide the organization and dedication which is necessary to produce this timely, thorough Proceedings of our 29th International Conference. Our thanks to all of you who made it possible.

Frank Roberts
President

ADCIS gratefully acknowledges the cooperation of Western Washington University Art Department in the design of the cover for the Conference Program and Proceedings for the last three years. This year's cover was done by Al Yates under the direction of Art Chantry.

ADCIS BOARD OF DIRECTORS

President
Frank Roberts
Amherst Wilder Foundation

President Elect
Glenn Head
Instructional Communications

Vice President
Robert Fratini
AT & T Technologies

Vice President Elect
G. David Peters
University of Illinois

Program Chair
Dale Brubeck
The Ohio State University

Presession Chair
Mary Trainor
Los Alamos National Laboratory

Special Interest Group Council Chair
W. Kirk Richardson
Georgia State University

Special Interest Group Representative
Donna Larson
Grand Valley State Colleges

Newsletter Editor
Tom Reeves
University of Georgia

Chapter Representative
Aaron Navarro
The Mitre Corporation

Executive Director
Gordon Hayes
ADCIS International Headquarters
Ex-Officio

SPECIAL INTEREST GROUP COUNCIL

Computer-Based Training
John Buck
Technassociates

Elementary/Secondary/Junior College
Gerald Blumenfeld
The University of Akron

Educators for the Handicapped
Delia Neuman
University of Maryland

Health Education
Donna Larson
Grand Valley State Colleges

Home Economics
Elaine Muller
Computer Education Consultant

Interactive Video/Audio
Bill Aggen
MetaMedia

Management Issues
Sue Leslie
IBM

Music
Charles Boody
Hopkins Public Schools

PILOT
W. Kirk Richardson
Georgia State University

PLATO
Celia Kraatz
University of Illinois

Telecommunications
Tom Downey
Boeing Aerospace

Theory and Research
David W. Dalton
University of Indiana

TABLE OF CONTENTS

COMPUTER BASED TRAINING (CBT)

Computerized Environment for Instructional Design: Developing CBT with Artificial Intelligence-Based Workstation 287
Avshalom Aderet
An Integrated CBI Authoring and Hardware System 1
Ann Brebner
Toward an Ideal Visual Interface for Computer-Based Authoring Systems 282
Dan G. Friedrich
Journaling Systems: Integrating Computer-Based Learning Activities with Applications Software 7
Shirley F. Gresham, Leslie R. Ferguson, Brockenbrough S. Allen
Introducing CHEMLEARN: Microcomputer-Based Training for CHEMLINE 14
Miriam L. Perkins, Melvin L. Spann, Michelle M. Geasler
What's Your CAI MO - Mainframe or Micro? 18
Barbara Hershey Handler
Training Videotape Production as a Means of Student Consultant Training 23
Joan Lauer Hayes
The Impact of Training in Enlightened Countries 30
John J. Hirschbuhl
Hidden Decisions in the CBT Design Process 37
Adele Suchinsky, David Losleben
Designing Software for Easy Translation Into Other Languages 41
Stanley R. Trollip, Gary Brown
Development of a Micro-Computer Based Knowledge System: An Overview 45
Alan D. White
On-Line Tools for 3-D Graphics Programming 50
Mason Woo

ELEMENTARY, SECONDARY, JUNIOR COLLEGE EDUCATORS (ELSECJC)

Teachers' Perceptions of Needs Related to Implementing Computer-Based Instruction 55
David W. Dalton
Computers in Education: What's Really Keeping Them Out? 60
Michael J. Hannafin, David W. Dalton, Simon R. Hooper
The Effects of Individualistic Versus Cooperative Computer-Assisted Instruction on Student Performance and Attitudes 62
David W. Dalton, Michael J. Hannafin
Using Computer Simulated Base Ten Blocks to Teach Double-Digit Addition and Subtraction 65
Gary A. Deatsman
Effects of Unit Test Item Feedback on Final Exam Scores for Computer Administered Exams 70
D. L. Jelden
The Computer Environment and the Teaching of Writing 75
Pam McRae, Mark Sherman
Technology and Group Instruction: A Communication and Management Tool for Teachers 78
Steven L. Robinson
Young Children and Computers: A LOGO Project Revisited 83
Charlotte Scherer
Teaching Modular Hierarchical Programming Using LogoWriter™ 88
Sylvia A. Shafto, John D. Lawson

EDUCATORS OF THE HANDICAPPED (SIGHAN)

Computer Applications for Transition Programs 96
Joseph P. DeMarsh, Pamela J. Zrihen

TABLE OF CONTENTS *(cont.)*

HEALTH EDUCATION (HESIG)

Developing a Knowledge-Based Simulation for Microcomputers 101
Thomas McFarland, Mary McFarland, Stephen Ragan, Diana Kottkey
An Evaluation System in Dental Hygiene Clinical Education 107
Michael W. Sachon, Patricia-Damon Johnson

HOME ECONOMICS CONSORTIUM (HOMEC)

Innovative Use of Plato Notefile in Dietetic Clinical Simulation 117
Leta P. Aljadir, Evelyn V. Stevens
Implementation of a Three Dimensional Animation Module in Designers Spatial Skills Educational Process 121
Susan Harshbarger
Utilizing an Interactive Computer-Based Approach for Enhancing the Study of Family Financial Planning 124
James L. Morrison
Addressing the Computer Literacy Needs of Home Economics Educators in Hawaii 127
Cecelia Thompson
What Can I Do With a Videodisc? 129
Clo Wiltse

IMPLEMENTATION (ISIG)

Human Related Factors in Implementing a Management Information System 132
Patricia V. D'Souza
Test Design for Computer Managed Instruction Systems 135
Robert L. Jorczak, Franklin C. Roberts
LADAR: A LAnguage for Describing Anticipated Responses in CAI Answer Judging 142
Hiroshi Kato, Yasuhiro Morimoto
Space Requirements for Computer-Assisted Instruction 147
Frederick G. Knirk, Daniel Christinaz
A Naturalistic Framework for the Development of a Computer Ethics 156
Donald Scherer
Bar Code Applications in an Academic Environment 163
Charles A. Schneebeck, Co-Nghiep Ho
The Application of Technology to a Basic Photography Class 170
Merton E. Thompson

INTERACTIVE VIDEO-AUDIO (SIGIVA)

Is There an Expert System in Your IV/A Future? 174
Greg Baur
Using Patient Simulations to Teach Problem-Solving 185
Lynn Johnson, Michael W. Finkelstein
Design Parameters for Video/Graphics Overlay Applications 191
D. V. Pigford

MINI/MICRO COMPUTER USERS (MINI/MICRO)

Statistics From the Top Down 196
Susan Athey
Authoring System Evaluation Package 201
Emile Attala, Marsha Epstein
An Evaluation of the Teaching Effectiveness of a PC—Based Emulation Program for Production Courses 207
K. Kern Kwong, C. Hirayavasit, S. Moon
Simulating an Army Aviator Qualification Course: Resource Allocation Model 210
Carol A. W. Sprunger, Paul J. Tremont
A Computer Augmented Curriculum 218
B. Wayne Walters

TABLE OF CONTENTS *(cont.)*

MUSIC EDUCATORS CONSORTIUM (MUSIC)

Improving the Measurement of Musical Ability Through Adaptive Testing 221
Walter P. Vispoel

PILOT

Origins and Early Development of PILOT 226
John A. Starkweather

PLATO USERS GROUP (PUG)

Generative CAI for Latin: Assessing a Ten-Year Project 229
Gerald R. Culley

THEORY AND RESEARCH (SIGTAR)

Integrating Problem Solving and Tutorial Strategies: Extending the Genetic Graph 234
Garland Bayley, William S. Bregar, Arthur M. Farley
An Electronic Coaching System to Teach Concepts of Personal Causation and Symbol Use to Severely Handicapped Learners 239
Gaye V. Fedorchak, Steven E. Nantz
A Cognitive Method for Designing Computer-Based Training 245
Veli-Pekka Liflander
Principles of Feedback and Related Variables in Computer-Assisted Instruction 253
Joseph S. Mattoon
SHUFFLE: An Intelligent System for Teaching Procedural Knowledge 256
Walter Maner
The Effects of Textual, Graphic and Animated Graphic Elaboration Strategies and Practice on Learning from Computer-Based Instruction 263
Lloyd P. Rieber
Online Training for Trainers: A Working Model Based on UNIX* 268
John A. Scigliano, Barry A. Centini, Jacques Levin, Alan White
*Unix is a Trademark of AT&T Bell Laboratories
Children Using Educational Software: An Observational Study 271
Rosemary E. Sutton
Measurement of Spatial-Symbolic Information Processing Skills 276
Arthur L. White, Donna F. Berlin

200 WORD ABSTRACTS (No Formal Papers)

COMPUTER BASED TRAINING (CBT)

Using a Workbook/On-Line Tutorial as a Starting Point for CBT Development with an Authoring Language 291
Martha R. Atwater
The Development of an Instructional Planner for an Intelligent Tutoring System 292
Melissa Berkowitz, David Emme, Stuart Macmillan
Computer-Generated Video Slides and Storyboards: Alternatives to Computer-Based Training 293
David W. Dalton, David Goodrum
Authoring System Evaluation Strategies 294
Daniel V. Eastmond
Adapting Object-Oriented Database Systems for Use as Automated Job-Aids 295
Shirley Gresham, Leslie Ferguson, Brockenbrough S. Allen
Software Tools for Training Developers 296
Greg Kearsley
Audio-Based CBT 297
James M. Kraatz, Jolanta Percy
Model Training Program for Reserve Component: A Pilot Study 298
Mahima R. Kundu

TABLE OF CONTENTS *(cont.)*

SIMUEDIT: Courseware Authoring Tool 299
Dave Lohn-Wiley, Zandria Merrill
Designing a Computer-Based Criterion Referenced Test 300
Francine Manna
Courseware Development for the Insurance Industry 301
John A. Merrill
Training the Microcomputer Illiterate to be Productive Users of the Technology in a One-Semester Experience 302
Richard L. Patterson
Learner Control: Where Do We Go From Here? 303
Joellyn Pollock, Cindy Leshin
Problems and Strategies for Developing CBI to Mirror a Mainframe System 304
Mary Riley, Shelley L. Morse
Corporate-Wide Computer Managed Instruction (CMI) 305
Joan Rogalska Saz, David F. Stevens
Computer Based Training (CBT) Cost Estimating Algorithm for Courseware (CEAC) 306
Ralph E. Schooley
Corporate Challenge: Computer-Based Instructional Gaming 307
Linda Shatzer
Checking Vendor Support 308
Timothy W. Spannaus
Learning Styles, Computer Attitudes and Preference for Instructional Delivery Methods 309
Charles J. Stanley, John C. LeDoux
An Authoring System Selection Guide 310
David F. Stevens, Joan Rogalska Saz, Marjorie Pomper

ELEMENTARY, SECONDARY, JUNIOR COLLEGE EDUCATORS (ELSECJC)

Interactive Video on a Shoestring — Revisited 311
L. Boyer, L. West
A Features Comparison of Print Shop for the Apple II, Macintosh and IBM 312
B. Clark, W. Clark, N. Higgins
Effectiveness of Writing to Read Program in Phoenix Public Schools in Arizona 313
Patricia Alba Mulligan
Computer-Based Testing in Schools 314
James R. Okey, Chris L. Peters, William H. Brittain
High Tech's Real Place - What Will It Take? 315
Dan Schwartz
Implementing an Applications Program Database 316
Scott D. Shoemaker, Norman C. Higgins
Validation of a Computer Administered Rhythm Tapping Skill Test for Elementary School Students 317
Aaron L. Spaulding, Gerald J. Blumenfeld, Ted Malcom
The Effectiveness of Using Computers in Developmental Writing Classes 318
Gary Vaughn
Electronic Support Networks for Improving Preservice and Inservice Teacher Training 319
Michael Waugh, James A. Levin
Effective Use of PLATO in the K-12 School System 320
Tom Williams, Marieanne Perrault, Mike Pineda, Lee Becea
Computer Based Education in California Public Schools 321
Pamela A. Wright

EDUCATORS OF THE HANDICAPPED (SIGHAN)

Educating Special Learners: A College Course 322
G. Phillip Cartwright
Computer Projects and Products for Hearing Impaired Populations 323
Phil Mackall

TABLE OF CONTENTS *(cont.)*

Using Cooperative Learning Teams in the Resource Specialist Program with Computers 324
Eleanor G. Thompson

HEALTH EDUCATION (HESIG)

Learning Behaviors of Baccalaureate Nursing Students and Computer-Based Education 325
Jean Arnold, Gail Iglesias
Evaluation Interactive Training for Acute Trauma Life Support 326
Mary R. Marlino, Thomas C. Reeves, Joseph V. Henderson
The Hurting Tooth: An Expert System to Diagnose Tooth Pain 328
Monica Nagy, Ernst Schelb

HOME ECONOMICS CONSORTIUM (HOMEC)

HELP: Development of a Home Economics Microcomputer Date Base System 329
I. Hathaway, M. L. McPherson, R. Heatley
Computer Applications in Family Resource Management 330
Arlene Holyoak
Seek and Find It With a Data Base 331
Dorothy Jensen
Home Economics Network: Progress and Promise 332
Ann B. Kapp
Computers in Home Economics: Graduate Level Teacher Education 333
Ann S. McChesney, Lynne Levy, Elaine Muller
Innovations in Computer Aids: The Development of a Quick-Reference Manual and a Slide Training Program 334
Phyllis Bell Miller, Jacquelyn DeJonge
Application of an Electronic Spreadsheet to a Foods/Foodservice Curriculum 335
Barbara F. Quinby, John Cheyer

INTERACTIVE VIDEO-AUDIO (SIGIVA)

Essential Arithmetic Skills via Interactive Video 336
Edward C. Beardslee
Interactive Videodisc Programs to Learn to Recognize Japanese 337
Alfred Bork
Instructional Engineering: A Place for a New Science 338
Richard Brandt
The Texas Learning Technology Group Complete Curriculum for Physical Science: A Pilot Status Report 339
Stephen F. Brown
Design Strategies for Interactive Videodisc Programs 340
Geri Gay
Basic Principles of Intravenous Therapy - An Interactive Video 341
Kathleen L. Guyett
C.A.S.T. - A Level IV Sonar Trainer 342
Tom Jones
CAST: An Interactive Videodisc System for Science Teachers 343
James M. King, Thomas C. Reeves, James R. Okey
Repurposing Videodiscs: A Research Study Examines the Issues 344
Carolyn S. Lay
Developing a Treatment for Content: An Example Using the Infowindow System 345
Mary Leight, Mary Richter
ARTsearch Videodisc Retrieval System for the Instruction of Art and Design 346
Patricia K. Mansfield
Effects of Computer-Assisted Video Instruction on Nursing Student Achievement and Learner Characteristics 347
Barbara J. McNeil

TABLE OF CONTENTS *(cont.)*

Developing CBE Software on the IBM Infowindow System 348
George A. Reed
Using Interactive Video to Assist Readers Comprehend Text 349
David Reinking, Jiin-Tian Chyou
Research Designs for Interactive Video 350
Dent M. Rhodes
The Video in Interactive Video: Making Pictures Speak 351
Janise Richards
Interactive Reading 352
Barbara Signer, Marie Gannon
Making the Media the Message 353
Lynne Stevenson
Scripting Interactive Video Lessons: An Approach that Meets the Needs of Both the Designer and Programmer 354
Mary Stoddard Trainor, Andrew E. Andrews, Stephen J. Long
Visual Considerations for Screen Design 355
Ralph Wileman

MANAGEMENT ISSUES (MISIG)

Faculty Incentives and Rewards for Development of CAI 356
Ronald C. Comer
An Evaluation of a Computer Curricular Integration Program 357
Elisabeth M. Craig
Computer-Based Instructional Design: A Few Solutions 358
Carol A. Dwyer
Why Instructional Software Projects "Fail" 359
Margret Hazen
Copyright Infringement: "Look and Feel" Implications for Software Developers 360
Thomas Lee Hazen
Planning for Integrated Technology-Based Instructional Centers 361
Joy R. Hughes
Strategies for Facilitating Faculty Acceptance of Technology Assisted Instruction
(Or "You Can Lead an Old Horse to Water But Can You Get Him to Use a Computer?) 362
Harvey L. Hughett
Forms, Forms, Forms: Aids for the Instructional Designer's Nightmare - Management 363
Marilynne W. Mathias
The CBT Design Document 364
Marjorie Pomper, Joan Saz
A Small IVA Development Project 365
Mary Jac M. Reed
Managing Computer-Based Instructional Systems: A Systems Approach 366
Debra E. Schweitzer

MINI/MICRO COMPUTER USERS (MINI/MICRO)

Computer-Based and Computer-Assisted Instruction in the Humanities: A Comparison of Techniques 367
John C. Fisher
The Design and Development of a PC-Based MAPICS Emulation Program 368
K. Kern Kwong, Donald Cheng, Wing M. Fok

MUSIC EDUCATORS CONSORTIUM (MUSIC)

The Role of the Computer in a Music Theory Classroom 369
Liora Bresler
Computer Simulated Rhythmical Training 370
Jon-Chao Hong, Der-Shen Cheng
An Intelligent Tutoring System in Harmony 371
Linda B. Sorisio

TABLE OF CONTENTS *(cont.)*

PILOT USERS GROUP (PILOT)

Using Special Graphic Effects in Instruction 372
Susanne M. Behe, Bradley K. Schaublin, Norman C. Higgins
Integration of CD-ROM and Optical Videodisc via PC PILOT in a Medical Application Program 374
Richard J. O'Connor
Souped-Up SuperPILOT Programming 375
Marianne E. Yoder

PLATO USERS GROUP (PUG)

U.S. Army Need for PLATO-Delivered Courses 376
Clinton L. Anderson
A Report on the Development of a State-Wide CBE Network in Maine Using NovaNET℠ 377
Owen F. Gaede
Teaching Exotic Languages with PLATO 378
Celia D. Kraatz, James M. Kraatz
CERL's "Colossal" Computer-Based Education System 379
Sam A. Falk Milosevich
The Effect of Immediate and Delayed Feedback on Retention in a Computer-Based Training Program 380
Susan D. Moisey
Development of PLATO Usage Among Faculty 381
Deborah Ouellette

THEORY AND RESEARCH (SIGTAR)

The Use of Knowledge-Based Microworlds in an Intelligent Tutoring System for Physics 382
Steven F. Brown
Simulation in Program Management Education 383
Forrest C. Gale
Feedback and Transfer with an Instructional Expert System 384
Scott Grabinger, Joellyn Pollock
The Effects of Ability-Based Grouping Strategies on Learning from the Computer 385
Simon Hooper, Michael J. Hannafin
Applying AI to Instruction 386
Greg Kearsley
Conditions Affecting the Effectiveness of Animated and Non-Animated Displays in Computer-Based Instruction 387
Dan Klein
Toward the Redefinition of Knowledge as Design: Implications for the Development of Computer-Based Learning Environments 388
Vincent E. Lasnik, Frederick C. Whiteman
The Computerized Advisory Service for Horticulture (CASH): Educating the Agricultural Community in the Field 389
Vincent E. Lasnik
Computer-Based Precollege Economic Education: An Imperative Need for Courseware Research and Development 390
Vincent E. Lasnik
Evaluating Courseware for Instructional Effectiveness 391
Cindy B. Leshin
Language and Thinking: A Computer-Mediated Critical Thinking Course 392
Denise E. Murray
The Computer as "Persona" in CBI 393
Henrietta N. Shirk
When CBT Is the Choice But Not the Solution 394
John Vukelich

AN INTEGRATED CBI AUTHORING AND HARDWARE SYSTEM

Ann Brebner

University of Calgary
Calgary Alberta Canada

ABSTRACT

An account is given of an integrated hardware and software system which provides easy-to-use facilities for a courseware author, and high quality multi-media presentations to a learner.

A hardware system has been developed for the production and delivery of such courseware. It may be based upon a variety of PC-DOS and MS-DOS microcomputers, using a super EGA graphics card with a resolution of 640 x 480 pixels and 16 colors from a palette of 64. An extra card is used for digitized speech, and for analog input.

An authoring system has also been designed to enable authors to make full use of this hardware. It provides the standard character set together with a resident set of mathematical and Greek characters; and there is a procedure for the creation and storage of custom-built characters. Other procedures allow an author to create graphic displays in a similar manner to that of a CAD system. In addition to the usual procedures for the display of text and processing of answers there are routines which allow authors to request and process positional input in the form of either points or lines and to use digitized speech.

This integrated system is easy to use and allows authors either to employ templates or to follow their own instructional paradigms. It allows the incorporation of expert tutorial programs. Since it is written in C, it is readily portable and has the capacity for modification and expansion.

INTRODUCTION

The predictions made in the 1960s concerning the development and use of computer based education have not been borne out. While there are many factors which have contributed to its slow growth, two significant ones relating to hardware and software can be readily identified. These are the lack of an effective, low-cost, multi-media delivery system that can be used in a wide range of computing environments; and the lack of any authoring environment in which good multi-media courseware may be developed quickly, at reasonable cost, and by instructors rather than by programmers.

The importance of these two factors has long been recognized. In co-operation with the National Research Council in Ottawa, the Faculty of Education Computer Applications Unit at the University of Calgary developed a number of CBT terminal systems for empirical evaluation. These systems were used at various educational levels and results have been reported from a number of controlled studies (eg. Taerum, 1974; Eaton, 1975).

The conceptual design of such equipment was gradually developed further, and was implemented on one of the earliest microcomputers in 1978. Systems of this type were used extensively for CBT with developmentally handicapped students (eg. Clark, 1980; Faulkner, 1982). However, they represent an older technology which is now essentially obsolete. Further, like all previous systems, they have limited flexibility. It would be difficult to replicate both the hardware and the associated software for use in other environments.

CBT courseware is used in a variety of computing environments. Sometimes, a program is used by a single learner with self evaluation of test results, and in this case a stand alone system is adequate. On other occasions, a employee of an organization, such as a bank teller or a ticketing agent, is required to upgrade job skills to a given criterion, and a workstation attached to a remote host machine is appropriate. In many educational situations, a group of students is required to use a program, with each working on a different unit, so that workstations attached to a multi-user system or local area network are needed, with student results collected on to one file handling device. In summary, a CBT delivery system is needed which will operate in a variety of computing environments.

Such a delivery system also requires an associated authoring environment, which allows courseware authors to develop programs inexpensively, using all the multi-media facilities of the workstation, and without requiring that they first become competent programmers. Further, the authoring system should also ensure a reasonable degree of portability by producing programs in a commonly used base language such as C.

A point has now been reached where the development of the technology makes it possible to resolve these two problems that have been inhibiting the development of CBT. A new CBT workstation has, therefore, been devised which is inexpensive, multi-media, and adaptable to a variety of computing environments. Also, an authoring system has been written for this workstation to meet the perceived needs of courseware developers, allowing them to develop quality lesson materials at reasonable cost.

THE MULTI-MEDIA WORKSTATION

The workstation, named CAITER, is based upon any IBM-PC or compatible machine with a mimimum of 640K bytes of memory and at least one empty slot. In addition it must have a super EGA graphics card, giving 640 x 480 pixel resolution with 16 colors out of a palette of 64; and at least one disk drive, although preferably two, either 5 1/4 or 3 1/2 inch.

In order to take full advantage of such display facilities, an NEC MultiSync monitor is used.

One card is added to this machine. It contains the chips and circuitry to facilitate the creation of digitized speech, and to support a pointing device such as a mouse or joystick.

Two character sets are built intc the system: a standard set of characters, and a set of Greek and mathematical symbols. Both use an 8 x 16 pixel font, and 25 lines of 80 characters. Other symbols may easily be created using author support programs. These may use the same font, or multiple blocks of 8 x 16 pixels. Any character may be rotated, placed anywhere on the screen, and magnified up to 15 times. These facilities enable authors of mathematical and foreign language programs, for example, to use the exact characters they would use on paper or on the chalk board. Also, the characters may be placed exactly where they are required, for example as superscripts or subscripts, vertically as labels on the axis of a graph, or diagonally on a map. From the user's standpoint, this eliminates the problem of constantly translating between arbitrary symbols, used because of the limitations of the CBT workstation, and the correct symbols used elsewhere. There is little doubt that the lack of such facilities, which are now offered by CAITER, has inhibited effective learning from CBT programs to date.

The CAITER system allows the display of lines from one to 15 pixels in width, and in any color. Closed curves on the screen may be filled, again using any of the colors. Moreover, any graphics developed using these procedures may be overlaid with characters from any of the text fonts.

The display and other output features of CAITER are designed to meet the needs of a learner. The objective is to exploit present developments in the technology so that unnecessary obstacles are not encountered when using a CBT program; to make use of graphical and pictorial images so that abstract ideas may be presented in more concrete form; and to make use of more than a single sensory modality in order to reinforce learning.

It is intended to include facilities to display digitized images in the near future. For this feature it is probable that 256 colors will be required and it will, therefore, be necessary to install an INTEL 82786 graphics coprocessor chip, or hardware which will carry out comparable functions.

Again, the needs of the learner are considered in relation to the input devices.

A normal keyboard is used for input of the standard character set, and may be covered by a template for input of the Greek and mathematical symbols or for author defined symbols. However, special purpose keyboards may be attached through an eight bit parallel input port. On a "concept" keyboard of this kind, the keys may be symbols, pictures, or even small objects such electronic components or coins.

As with output, it is important and also quite practicable to avoid using arbitrary symbols, which impose an unnecessary load upon a student's memory and require repeated translations which inhibit learning. On the contrary, it is possible to make learning easier by taking it closer to the concrete level. In a program to teach laundry symbols to slow learners, for example, the keys can be marked with the actual symbols that are used by the manufacturers; and in programs concerned with money handling, the keys can be bills and coins.

Positional input is also available through the use of any analog device such as a joystick or a mouse. This form of input frequently makes it easier to interact with a program.

Positional input may take the form of selecting part of a text, graph, or picture; or it may require the student to track a line or create a drawing. For example, the student may be asked to mark on a map a route covered by early fur traders, or that taken by Lewis and Clark!

With these input and output facilities available on CAITER, the problem is next to design and implement an authoring environment which allows the courseware author to take full advantage of the facilities and to produce courseware reasonably quickly and inexpensively.

THE AUTHORING ENVIRONMEMT

Author Interface

For a long time it has been clear that among the major obstacles to the widespread acceptance and use of CBT are the shortage of quality courseware and the high cost of producing such courseware (Hallworth and Brebner, 1980). Further, there now appears to be emerging a consensus that there is needed not merely a CBT language but a CBT authoring environment (Merrill, 1985).

Almost without exception the present CBT languages attempt to simplify the writing of courseware, and consequently lower the costs, by restricting themselves to the use of one instructional paradigm, generally without the inclusion of images or speech, and frequently without the use of graphics. Lower cost is achieved by sacrificing flexibility and, ultimately, quality.

The particular instructional paradigm that is commonly used may be characterized as typical of instruction offered in schools produced by the Industrial Revolution. In certain respects, and particularly in training situations, it has been shown to be quite effective. However, it should incorporate the use of multi-media terminals; and a very brief consideration leads to the conclusion that this is far from being the only instructional paradigm used by experienced teachers, particularly in tutoring situations.

For use with the CAITER delivery system, therefore, an authoring environment has been designed which makes it possible to develop courseware expeditiously while still making use of multi-media output and a variety of input devices.

Further, it is assumed that courseware is not necessarily frame based. Authors are not required to build programs which conform to any specific instructional designs. Instead, the assumption is made that an experienced teacher, who knows the subject area, will already have in mind the strategies which work well in particular instructional situations and will wish to implement them in CBT courseware. However, in the event that an author wishes to make use of a particular instructional design, a template can be made available from a library.

From the author's point of view the system consists of a series of menus, arranged in tree structure, giving access to the author support routines which make the production of multi-media courseware comparatively easy for a non-programmer. Experience to date has shown that access in this manner, where choices on each menu are made by positioning a cursor with a joystick, is sufficient for all but the most experienced author, who can more readily use a set of terse menus and a facility to move more flexibly between them.

At the present time such an authoring system makes quite heavy demands upon the processing power and memory of the microcomputer. However, rapid technological developments are taking place in both these areas, and progress is such that with the increasing power of microcomputers and the size of memory they can address, microcomputers will soon have the power of mainframes. Indeed, it becomes difficult to foresee the place that mainframes will have in the educational computing world within five years. On the other hand, there is little doubt that special purpose microcomputers will proliferate, using special purpose software systems.

CAITER is one of these specialized microcomputer based systems.

Support Facilities

The authoring system allows access to programs which make it a relatively easy task to develop and edit displays, to accept and judge student responses, to control the flow of lesson material, and to keep a variety of records.

Characters on displays are produced using the standard or supplementary keyboard. In cases where these are not appropriate the author may access further routines which allow the creation of new characters by moving the cursor around an enlarged 8 x 16 font. Pixels are fixed in the font to make the new character which may then be edited, displayed in standard sizes both horizontally and vertically in conjunction with other characters, re-edited if necessary, and finally saved as part of a new character set. A character so constructed may be assigned to a specified key on the standard keyboard, or on an alternate keyboard.

Graphics may be entered into the courseware by drawing on the screen with the joystick. Using the appropriate menus, the author can access support facilities for assistance in drawing straight and curved lines, squares and rectangles, circles and ellipses, and arbitrary shapes. Shapes may be translated, scaled, or rotated; also, they may be shaded or colored. In effect, the system has many of the facilities found in a CAD system.

A display may contain both text, including author designed characters, and graphics; and there is full control of both foreground and background shading and color.

Further support facilities allow the author to create fully digitized speech, and to use it in a piece of courseware without difficulty.

In addition to fixed author designed displays, dynamic displays whose precise form is determined at run time may also form part of the courseware. Thus the exact form of a display may be determined by random generation of some items. Alternatively, it may be modified as a result of a single student response such as a mathematical function, or as a result of a history of previous student responses.

Procedures are also available through the menu which enable an author to obtain and judge student responses that are either symbolic or positional.

Symbolic input may be numeric or alphabetic, and in addition may contain any of the special symbols created by the author. Tolerance levels may be set for numeric input. Alphabetic input can be evaluated using the techniques found in current CBT languages. Also, student responses in the form of equations can be accepted and evaluated.

Positional inputs which can be evaluated are made by use of an analog device. These may be in the form of a point, within a preset tolerance, used for example to identify a feature on a map, or a specific point on a mathematical graph, or one of several choices on a menu or a multiple choice test. It may be a substring, used for example in courseware to identify grammatical parts of a sentence. However, it may also take the form of a line which the student draws, for instance, on a map or graph. The support programs make it easy for an author to specify tolerance limits in this case also.

There are facilities for the author to specify both initial and restart entry points in a program. For example, a student who fails to complete a lesson may be restarted next time at the beginning of the lesson, at the point where the student terminated, or at any point determined from the history of past performance.

As in most CBT languages, simple branching after judgement of a student answer is available to provide feedback. This allows for multi-state branching after a response which is correct, partially correct, a common error, or a repeated error, to give either reinforcement or a hint.

Larger scale branching, as required in several other circumstances, is also available. A student may be branched to an enrichment or remedial lesson following an analysis of performance. An instructor may intervene and branch a student to a different lesson. Further, student controlled branching may also be built into the courseware. The student can then return to earlier questions and answers or move ahead to new material if so desired.

Facilities allow for student records to be kept automatically, and data thus accumulated may be analyzed to provide histories of student performance or to evaluate the courseware itself. These data include not only responses to questions and the paths taken through the program, but also the time required to complete parts of the program and the total program.

COURSEWARE DESIGN

Taken as a whole, the support facilities described above allow the author access, through a series of menus, to a set of procedures which make it relatively easy to include in a lesson all the special features of the CAITER delivery system which the author deems appropriate for the particular topic area.

The author uses these procedures to create a teaching/learning unit. This may be considered as a relatively brief instructional transaction. It will not necessarily be the same as the traditional "frame". Rather, the system is designed to place the emphasis upon a lesson, as a sequence of units or instructional transactions, designed by an experienced instructor.

The courseware system does not require the author to follow or incorporate any particular instructional designs. It merely places the basic tools for multi-media instruction into the hands of an instructor who is already an expert in teaching a particular subject area.

However, some instructors will frequently use particular instructional designs, and will require access to them. There are, therefore, facilities for creating and extracting design templates. In addition, it will be possible to create libraries of those units which are commonly required.

Further, the authoring system is so constructed that it is possible to move into other languages in order to take advantage of routines for presenting specific instructional sequences. It is then easy to return to the CAITER authoring system, bringing any resultant data. It is also practicable to move between systems in this manner as frequently as needed in order to help the student to learn effectively. It is anticipated, for example, that routines in other languages will be needed to provide simulations, to help students perform calculations, to allow them to make use of expert systems, and to provide syntactic analysis and synthesis.

In effect, with an authoring system of this kind it is possible to produce a wide variety of lesson types. Clearly it is feasible to use the system to implement courseware developed on earlier systems. However, the facilities will also allow authors to work creatively to produce new forms of courseware.

Some courseware produced to date includes a physics unit on projectiles and one in mathematics on conic sections, both making use of the graphics facilities. A social studies unit uses the graphics to draw a map of Canada and positional input to indicate the routes taken by early explorers. Applications which utilize the graphics in conjunction with the audio facilities are the functioning of the heart, and aspects of a four stroke gas engine.

CONCLUSION

Computer based training has progressed a long way from the text bound courseware of the early mainframes. However, the fact that it has yet to make a significant impact in education, or even in business and industrial training, raises serious questions regarding the adequacy of present CBT systems.

It has been suggested that two important reasons for the comparative failure of CBT to date have been, first, the inadequacy of the display hardware presently available; and, second, the high cost of producing courseware.

The CAITER delivery system described in this paper, and the associated authoring system, are intended to provide a solution to these problems. The CAITER workstation is designed to meet the needs of the learner: it is multi-media, provides displays using the symbols that the learner is expected to use in real life situations, offers a variety of input devices from which the learner may choose that easiest for a given program, is easily portable, and can be used in most computing environments.

The authoring system is designed to help the author meet learner needs, whilst at the same time making courseware development both easier and quicker, and therefore less expensive. It is also designed to allow the use of a variety of teaching paradigms, including those considered most appropriate for tutoring an individual learner.

Together, the terminal hardware and the authoring system would appear to provide a solution to some of the most significant impediments to the widespread development of CBT. It is hoped they may indicate a direction for future developments.

REFERENCES

Clark, L.M. Vocational Training for the Retarded by CAI. Unpublished M.Sc. thesis, University of Calgary, 1980.

Eaton, P. Visual Discrimination and CAL. Unpublished Ph.D. thesis, University of Calgary, 1975.

Faulkner, M.C. Time Training for the Handicapped by CAI. Unpublished M.Sc. thesis, University of Calgary, 1982.

Hallworth, H.J. and Brebner, Ann. Computer Assisted Instruction in Schools. Edmonton: Alberta Education, 1980.

Merrill, M. David. Where is the Authoring in Authoring Systems? Journal of Computer-Based Instruction, 12 (4), Fall, 1985, pp. 90-96.

Taerum, T.V. Computer Assisted Instruction in Statistics for University Students. Unpublished Ph.D. thesis, University of Calgary, 1974.

Ann Brebner
The University of Calgary
2500 University Drive N.W.
Calgary, Alberta T2N 1N4
Canada

Journaling Systems: Integrating Computer-Based Learning Activities with Applications Software

Shirley F. Gresham, Leslie R. Ferguson, and Brockenbrough S. Allen

Department of Educational Technology
San Diego State University

ABSTRACT

Teaching people to use software is usually expensive and time consuming. Journaling can be a cost-effective alternative to extensive print tutorials or human assistance. Journaling software takes control of a featured application and simulates keyboard and mouse activities. This permits "the system" to operate the computer, leaving the learner free to observe and listen. Journaled programs can be synchronized with audiotape. The combination, (often called a guided tour) permits simultaneous presentation of overview information with examples.

This article focuses on two journaling programs for the Macintosh family: an "event-manager" developed by the Apple Computer Company and a desk accessory developed by Affinity Software. The article discusses techniques for developing guided tours and examines instructional design issues relating to pacing, learner control, testing, and formative evaluation.

Reading printed documentation and tutorials can be a frustrating way to learn new software. Learners often find it cumbersome to "walk-through" a software application when they themselves must coordinate the operation of the computer with off-line instructions and descriptions.

Non-print training materials can be expensive and time consuming to produce. Video-based training programs are hard to modify or update when software is changed: designs are not easily evaluated prior to actual production. Some applications software incorporates on-line tutorials. However, these do not allow for audio narration, and they cannot be easily adapted to a specific population of users.

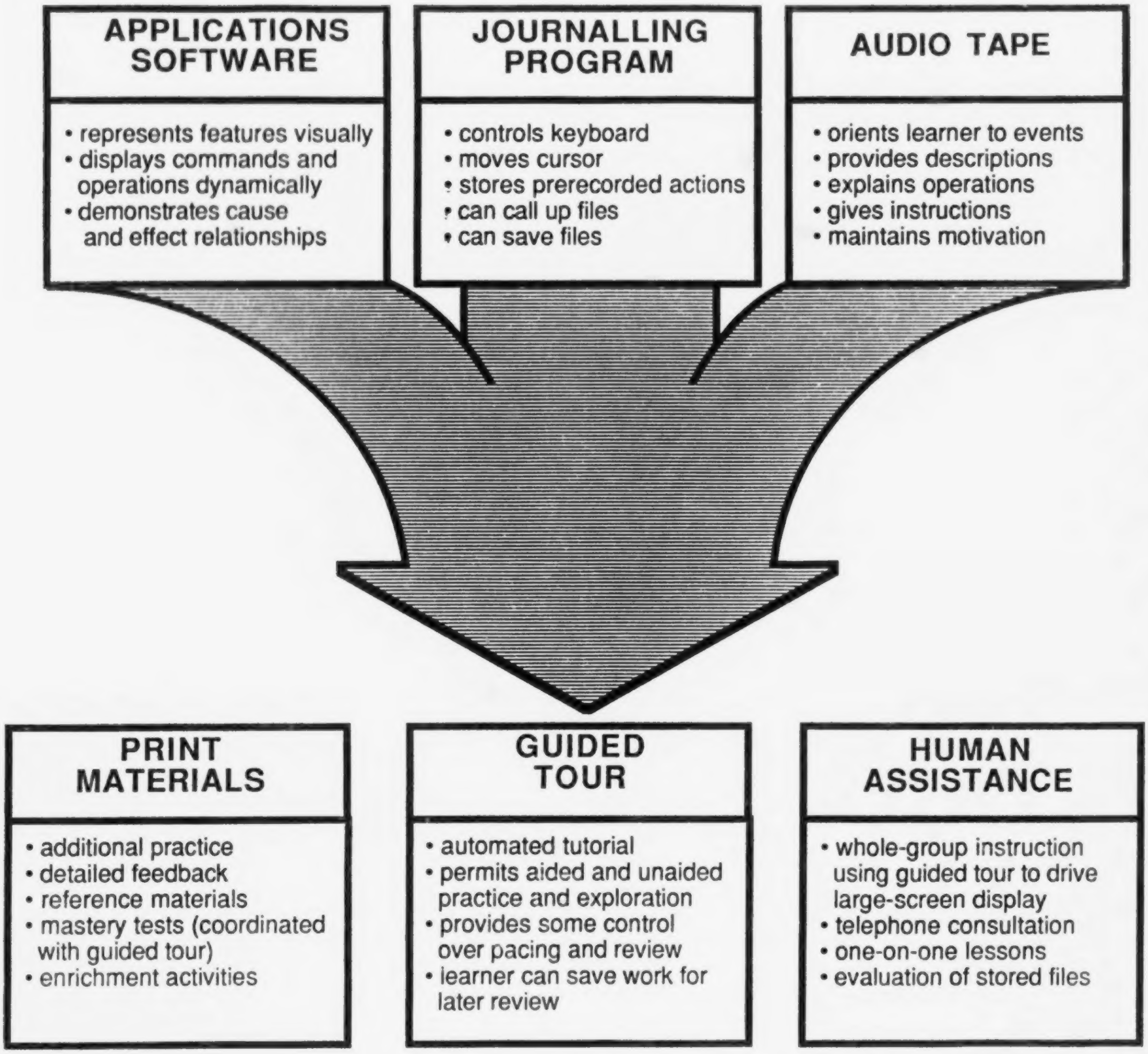
WHAT IS JOURNALING SOFTWARE?

Journaling software takes control of a featured application and simulates keyboard and mouse activities. This permits "the system" to operate the computer, leaving the learner free to observe and listen. Learners can engage in aided or un-aided practice at various points during the journaled program.

Although journaling software is available for many personal computers, this article will focus on two journaling programs for the Macintosh family. The first program, developed by the Apple Computer Company (1984, 1985), is driven by an "event manager" that records keyboard and mouse activities. The event manager replays the recorded activities in response to various menu options selected by the learner.

The second type of journaling program is macro-driven and is represented by Tempo ™, an inexpensive desk accessory developed by Affinity Software (1986). Tempo records a sequence of mouse moves and keystrokes. These can be recalled by name or by pressing designated key combinations. Other macro programs are available for the Macintosh and some applications come with their own systems for defining and saving macros.

© 1987, S. Gresham, L. Ferguson, & B. Allen



APPLICATIONS SOFTWARE

- represents features visually
- displays commands and operations dynamically
- demonstrates cause and effect relationships

JOURNALLING PROGRAM

- controls keyboard
- moves cursor
- stores prerecorded actions
- can call up files
- can save files

AUDIO TAPE

- orients learner to events
- provides descriptions
- explains operations
- gives instructions
- maintains motivation

PRINT MATERIALS

- additional practice
- detailed feedback
- reference materials
- mastery tests (coordinated with guided tour)
- enrichment activities

GUIDED TOUR

- automated tutorial
- permits aided and unaided practice and exploration
- provides some control over pacing and review
- learner can save work for later review

HUMAN ASSISTANCE

- whole-group instruction using guided tour to drive large-screen display
- telephone consultation
- one-on-one lessons
- evaluation of stored files

The applications software, a journalling program, and an audio tape combine to create a cost-effective training medium, which is often called a guided tour. These guided tours can be integrated with print tutorials or other training media to produce a comprehensive training package.

Guided Tours

Developing a guided tour is relatively simple. An informal program can be produced in a few hours. Systematically-designed programs require additional planning and resources.

To journal a program, the author performs keyboard and mouse operations in real time. The journaling software records these interactions for playback under system control.

A journaled program can be synchronized with an audiotape. The combination, (often called a guided tour) permits simultaneous presentation of overview information with examples.

The computer and the tape player are not physically connected; audio cues synchronize the playback. The user boots the program, waits until it is ready for a command, engages the tape player, and then presses a key or mouse button on hearing the cue.

Some training situations may not require audio narration. An alternative to the guided tour is to add insets to the screens of a file running under the featured application that describe the journaled actions. The author can use a desk accessory that pastes electronic "sticky notes" on screen displays when the featured application lacks a convenient way of generating text displays.

A Typical Guided Tour

1. Lessons can incorporate print materials. Such materials should include directions on how to use the tour.
2. The learner boots the guided tour disk. If the tour is macro-driven, the learner is taught how to call the macros before he or she begins the first lesson.
3. If the lesson is event-managed, the learner clicks on the appropriate menu button to begin a session. If the lesson is macro-driven, the learner activates macros from a menu or by keyboard commands.
4. After finishing a session, the learner can explore the capabilities presented or practice specific techniques.
5. The learner may then be directed to turn to a print package for further instructions about the featured software or for testing on lesson content.

EXAMPLES OF JOURNALING SOFTWARE

Apple Journaling Software for the Macintosh

Originally developed by Apple for authoring its own guided tours, this software is now available from a number of sources of public domain programs, including Macintosh user groups. When completed, a guided tour disk will contain:

- The journaling program—controls the computer portion of the guided tour.
- The featured application—software controlled by the guided tour.
- Scriptfiles—contains the guided tour menu screen session names, and the name of the featured application.
- Session files—contain a record of journaled activities (keystrokes and mouse moves) for each session or lesson.

This system is relatively easy to use. After filling in the Scriptfile with the application and file names, the author begins the journaling process by "shift-clicking" on a menu button associated with a specific session. The author then "walks-through" keyboard and mouse actions in real time, creating a sequence of screen activities that will eventually be viewed by the learner.

To check his or her work, the author clicks on the menu button. (To use the guided tour, an end user would do the same.) To ensure that nothing is recorded over the Script and Session files, the author changes the "DoesNotRecord" icon name to "Finder" when journaling is complete.

Journaled programs developed with Apple's software are transparent to the end-user: he or she need only insert a disk and turn on the computer to get to the guided tour menu screen. The learner proceeds through the guided tour by clicking on session buttons—each representing 5 to 15 minutes of recorded keyboard and mouse activity.

The learner may practice at any time between sessions by stopping the audiotape and clicking on the appropriate menu button (often labeled "Let Me Try [application name]"). Sessions can be repeated if the audiotape is re-synchronized.

Apple's software maintains an untouched version of lessons so they will look the same each time they are started. This is an advantage over macro-based journaling programs because the author does not have to deal with the problem of restoring the journaled file to its initial state.

Affinity's Tempo

This software is primarily designed to store routine or repetitive operations used with applications such as word processing and spreadsheets. However, it is easily adapted for use as a journaling program.

Tempo must be installed as a desk accessory.on the startup disk that will be used with the featured application. Selecting Tempo from the desk accessory menu installs the working Tempo menu on the menu bar. The author creates new lessons by selecting the Record option.

To create a macro, the author defines it by building a sequence of keystrokes and mouse moves. The macro is then assigned a name. Hundreds of keystrokes or moves can be assigned to a single macro. To create a journaled program, the author defines a small library of macros which then appears on the Tempo menu. The end-user selects macros from the library in response to taped instructions. Macros can also be called by pressing a combination of keys previously defined by the author.

Macros work in two modes: real time and computer time. Real time is the actual time it takes to journal a session (and also the time it takes the learner to view the session as it is played back). A macro executed in computer time performs operations at maximum speed; screen events usually move too rapidly for learners to follow.

The computer-time mode can be used for specific functions that are not of interest to the learner, such as "clean-up" routines. Learners are instructed to activate these macros after completing a lesson. Pre-planned routines then restore session screens to their original state. There is no need for clean-up routines with Apple's journaling system because lessons are automatically refreshed.

With Apple's event-managed journaling system, users can stop only at the end of a session. In contrast, macro-based systems allow users to pause at any time during a session. The user can stop to handle other affairs by simultaneously depressing the mouse button and the pause button on the cassette player. Macro-driven journaling is more flexible and powerful than event-managed journaling but has disadvantages that will be discussed later in this article.

INSTRUCTIONAL IMPLICATIONS

Journaling is a simple yet powerful alternative to authoring systems and languages as a means for developing concurrent software training. Although capabilities are limited, careful design can result in elegant solutions.

Instructional Design

Journaling systems can simultaneously describe and depict. Combined use of audiotape with computer displays enables learners to listen to general statements about an application while they see examples on the screen. This is arguably one of the most important instructional capabilities of any audiovisual system: that it allows coordination of generalities (audio-based descriptions) with specific examples (visual depictions).

The efficacy of various combinations of audio and visual presentations has been a consistently controversial issue for several decades. But in general, the literature argues for careful attention to the way in which audio information is integrated with visual information. Fleming and Levie (1978) proposed several principles of instructional message design relating to "two-channel" (audiovisual) instructional systems:

- where an audiovisual presentation is too rapid, the perceiver must choose between the two channels (p.60);
- capacity appears to be larger where two modalities are utilized (audition and vision) rather than one (p. 61);
- Two tasks involving the visual modality. . . will interfere more than where one involves the visual and one the auditory modality (p. 61).

Journaling software relies on numerous "chunks" of information; specific messages are represented in the form of audio, computer, or print displays. The instructional design process can be enhanced by theories or models that help to identify the purpose or function of each component in the overall instructional strategy. M. D. Merrill's

(1983) Component Display Theory and Gagné's (1977) nine events of instruction are useful in laying out the basic structure of a guided tour.

Reviews and repeats of lesson contents can be built into lessons or provided through a recommendation that the learner view lessons a second time. Directions can be delivered after the session:

> You may want to go through this lesson a second time. To do this, rewind the cassette tape to the beginning of Lesson Two. If you have already done this, or you feel you have mastered this section, continue on. If you want to review, rewind the tape now.

When a lesson segment does not start at the beginning of a tape, it may be difficult for the learner to find the starting point. A simple solution is to do some "cassette housekeeping" at the beginning of each lesson:

> Before we begin, let's do a little cassette housekeeping. Find the counter on your cassette player and reset the counter number to zero. (pause) This will help you find the beginning of this lesson for subsequent lesson reviews.

Learner-Driven Activities

Practice improves comprehension, creates an active role for the learner, and lays the groundwork for independent use of the featured software. Practice sessions should be carried out on a separate disk to avoid contaminating other files. Content from the main disk can be transferred to the practice disk if necessary.

Specific instructions for practice can be included on the audio tape. An alternative approach is to direct the learner to printed guidance:

> Now you're ready to practice adding and deleting records. Directions for closing the current disk and for booting up the practice disk are on page eight of your print package. The flowchart on page nine will help you remember the steps. Pause the tape and read page eight now.

Journaling permits descriptions of complex operations to be integrated with actions as they are performed by the learner. Following a macro-controlled segment, a journaled lesson automatically returns the learner to complete active control of the featured software. This makes it easy to insert learner-driven segments between macro-based presentations.

The audio-tape can provide step-by-step coaching as learners attempt the operations. Learner-driven activity can be used to move a macro-based tour to other screens or files, thus setting the stage for the next journaled lesson:

> Find the graphic on this screen labeled "Computer Software Department." Place the mouse arrow directly in the middle of this graphic. When you hear the third tone, double click the graphic. [Three tones] You have just performed what is called a "link". A link is used to open another file without going to the file menu.

Integrating Delivery Systems

Guided tours can be coordinated with other training materials. Vendor documentation often omits important concepts or techniques; supplementary print packages can expand on the original documentation and adapt it to the needs of specific groups of learners. Job performance aids for often-used procedures can also be included and used in conjunction with the audiotape. Print supplements for a macro-based guided tour should include instructions on how to activate the macros.

Mastery Testing

Mastery test items can be integrated into the guided tour format or coordinated with the supplementary print package. Instructions on the audio track tell the learner when to take the test and where to find relevant print information:

> Now that you have completed Lesson Three, turn to page ten of your guide and follow instructions for the section titled "A Test on Lesson Three." You may use the flowchart on page nine of the guide to assist you. Pause the cassette tape and take the test on page ten now.

Test items can take a variety of forms. The simplest approach is to have the student take an off-line printed test as suggested in the excerpt above. Off-line tests can contain printed "dumps" of relevant screen displays.

Tests can also be included in the guided tour itself. This is more practical with macro-based tours, since the student can stop the audiotape at any time and take active control of the featured application. The student can save or print a file of his responses for subsequent review by an instructor or supervisor.

An alternative testing technique for use with event-managed journaling programs is to use the audiotape to narrate the test. A pause is inserted in the tape so that students will have time to write their response on a printed test form.

DEVELOPING GUIDED TOURS

Quick Authoring

One technique for authors who have prior experience explaining the featured application is to write a rough outline of the narration. The author then records his or her comments while simultaneously journaling the lesson. This helps to ensure appropriate pacing. Quick authoring is especially appropriate when the target audience is small, when the journaled program will have a limited shelf-life, or when time and resources are limited.

Systematic Design

More sophisticated approaches to authoring are based on systematic analysis of subject matter and learner needs, explicit statements of instructional objectives, carefully planned instructional strategies, and planned integration of print-, audio-, and computer-based displays.

Careful front-end analysis establishes a firm foundation for scripting. One useful format is the standard two-column structure used in the development of many audiovisual productions. The audio column contains the narration and the visual column contains the corresponding screen actions. Storyboards with mockups or print-outs of screen displays may also be helpful.

The recorded narration should be conversational, informal, and should use the active tense. Using contractions and other natural speech patterns makes listening easier. In most cases, the author should explain the parts of the screen the learner is seeing and explain new actions as they occur.

Timing is important. Journaling with a rough audio recording helps to establish the pacing of the presentation. Since the audio portion of a guided tour is synchronized to the operating speed of the computer, tapes developed for a particular model may not synchronize properly with faster or slower models. In addition, differences in disk-access time (e.g., floppy vs. hard disk) can affect the synchronicity of the tape and the journaled program. These problems can be minimized by avoiding routines that require extensive access to disk files and by loading the entire journaled program into a RAM cache.

Formative Evaluation

One of the advantages of journaled software is the comparative ease with which it can be subjected to one-to-one formative evaluation (Dick and Carey, 1985). One technique is to read the script as the learner works through the program. This avoids the need for synchronizing the narration exactly, since the reader can adjust the timing. A second observer can take detailed notes on learner interactions. If the journaled program is macro-based, the evaluator or student can stop the playback of the macro sequence at any time for a detailed debriefing.

After the formative evaluation, the final version of the audio track should be recorded using the best production quality available. The author then journals to this final narration.

SUMMARY

Journaling programs require little author training and can be adapted to a variety of needs and cost constraints. Guided tours are relatively easy to design and to evaluate and can be integrated with print tutorials or other training materials. The flexibility and cost-effectiveness of journaling makes it an attractive alternative for adapting software training to the needs of a particular organization or target population.

REFERENCES

Affinity Microsystems. (1986). Tempo, Version 1.1 (computer software). Boulder, CO.

Dick, W. & Carey, L. (1985). The systematic design of instruction. Glenview, IL: Scott Foresman.

Fleming, F. & Levie, W. H. (1978). Instructional message design. Englewood Cliffs, NJ: Educational Technology Publications.

Gagné, R. M. (1977). (3rd ed.). New York: Holt, Rinehart, and Winston.

Hasmig, S. (1985, March). A guide to making guided tours. (unpublished manuscript). Cupertino, CA: Apple Computer Co.

Merrill, M. D. (1983). Component Display Theory. In C. R. Reigeluth (Ed.), Instructional-design theories and models: An overview of their current status (pp. 279-333). Hillsdale, NJ: Lawrence Erlbaum Associates.

Brockenbrough S. Allen
San Diego State University
Dept of Educational Technology
San Diego, CA 92182-0311
(619) 265-5439

Introducing CHEMLEARN:
Microcomputer-Based Training for CHEMLINE

Miriam L. Perkins and Melvin L. Spann, Ph.D.
National Library of Medicine

Michelle M. Geasler
Online Computer Systems, Inc.

ABSTRACT

The National Library of Medicine (NLM) developed and maintains CHEMLINE (Chemical Dictionary Online), an online, interactive chemical dictionary file. NLM has developed CHEMLEARN, an interactive microcomputer-based training package designed to teach medical and health professionals, as well as librarians and information scientists, how to effectively use CHEMLINE.

INTRODUCTION

The National Library of Medicine in conjunction with Chemical Abstracts Service developed and maintains CHEMLINE (Chemical Dictionary Online) (1), an online, interactive chemical dictionary file. The file provides a mechanism whereby information on over 750,000 chemical substances can be searched and retrieved online. It contains CAS Registry Numbers, molecular formulas, CA chemical index nomenclature, generic and trivial names, classification codes, and a locator designation which points to other files in the NLM system and the Toxic Substances Control Act (TSCA) Inventory. In addition, where applicable, each record contains ring information including: number of component rings within a ring system, ring sizes, ring elemental compositions, and component line formulas. In an effort to provide widespread, easily accessible, inexpensive instruction in the use of CHEMLINE, NLM has developed CHEMLEARN, an interactive microcomputer-based training package designed to teach medical and health professionals, as well as librarians and information scientists, how to effectively use CHEMLINE. The microcomputer version of CHEMLEARN is based on a main-frame CAI program developed at the National Library of Medicine in 1979 (2).

OBJECTIVE

The objective of CHEMLEARN is to provide a fully comprehensive, interactive, microcomputer-based instruction package for CHEMLINE. CHEMLEARN's menu driven design allows the novice user to learn about the basic content and use of CHEMLINE, such as the retrieval of Registry Numbers and synonyms, while allowing the skilled searcher to learn about the more complex aspects of the database, such as substructure searching. CHEMLEARN may be used in place of formal training, as a precursor to or a refresher following formal training, or for review of a particular concept.

CONTENT

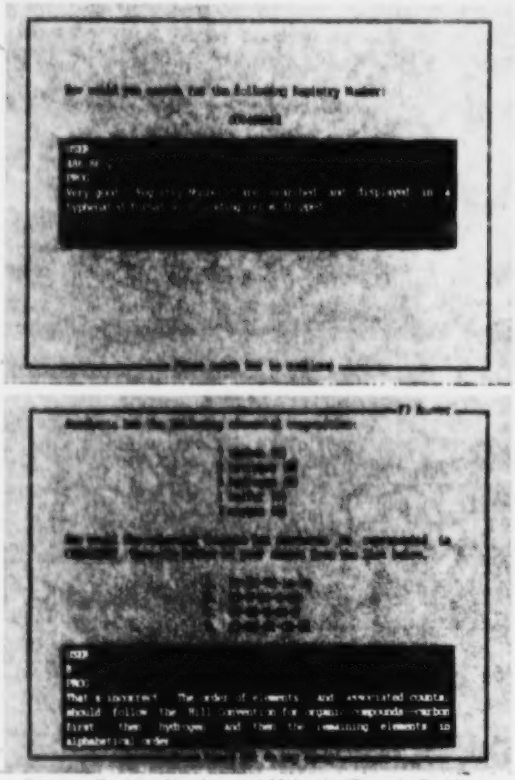
CHEMLEARN is organized into seven chapters. The introduction is presented in chapter one. Chapters two, four, and six contain discussions of the data elements in CHEMLINE, followed by multiple choice and/or fill-in-the-blank problems. Chapter two covers the basic data elements used when searching CHEMLINE. The fragmented data elements are covered in chapter four and the ring data elements in chapter six. Chapters three, five, and seven provide simulated searches which illustrate actual online sessions. It is recommended that new users proceed sequentially through the program. A hierarchical listing of each of the sections in CHEMLEARN is given below.

Introduction
Basic Data Elements
. Registry Number
. . Registry Number
. . Other Registry Number
. . Related Registry Number
. . Searching By Registry Number

  - Molecular Formula
    - Molecular Formula
    - Searching By Molecular Formula
    - Molecular Formula Truncation
  - CAS Type 1 Name
    - CAS Type 1 Name
    - Searching By Chemical Name
    - Chemical Name Truncation
  - Synonym
    - Synonym
    - Searching By Common Name
    - Searching By Chemical Name
    - Common Name Truncation
  - Locator
    - Locator
    - Searching By Locator
  - Classification Code
    - Classification Code
    - Searching By Classification Code
- Simulated Searches Utilizing Basic Data Elements
  - Caffeine Search
  - Tris Search
  - Dioxin Search
  - Chlorpromazine Search
- Fragmented Data Elements
  - Name Fragment
    - Name Fragment
    - Searching By Name Fragment
    - Name Fragment Truncation
  - Formula Fragment
    - Formula Fragment
    - Searching By Formula Fragment
- Simulated Searches Utilizing Fragmented Data Elements
  - Benzenesulfonamides Search
  - Benzodiazepines Search
- Ring Data Elements
  - Number of Rings
  - Ring Size
  - Ring Elemental Analysis
  - Component Line Formula
  - Searching By Ring Data Elements
- Simulated Searches Utilizing Ring Data Elements
  - Benzothiadiazines Search
  - Phenothiazines Search

INSTRUCTIONAL DESIGN

User interactions with CHEMLEARN (problems and simulated searches) are displayed in an area of the screen designed to simulate user-system interactions during actual CHEMLINE sessions. Users interact with CHEMLEARN by entering a letter, word, or phrase after the USER: prompt, followed by a carriage return. CHEMLEARN responses are displayed following the PROG: prompt. For purposes of illustration, user input, as well as search terms, appear in yellow. Figures I and II illustrate user-system interactions with CHEMLEARN.



```
USER:
B
PROG:
That's incorrect.  The order of elements, and associated counts,
should follow the Hill Convention for organic compounds—carbon
first, then hydrogen, and then the remaining elements in
alphabetical order
```

Figure I Registry Number and Molecular Formula Problems

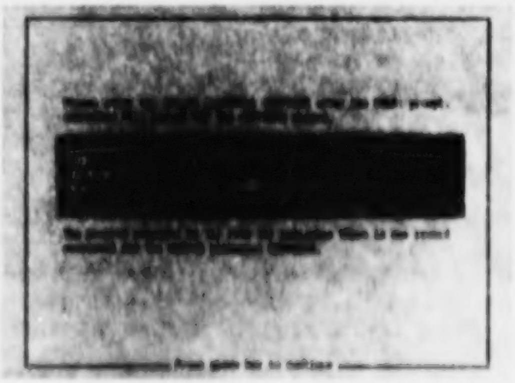
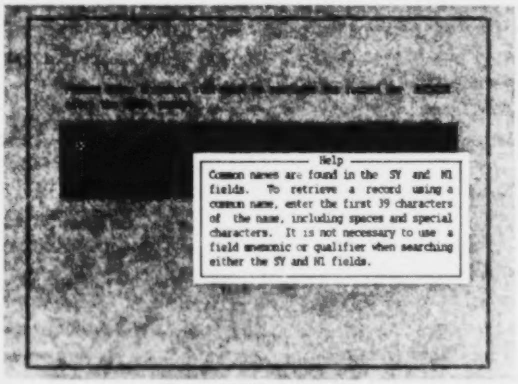


Figure II User-System Interaction from the Caffeine Simulated Search

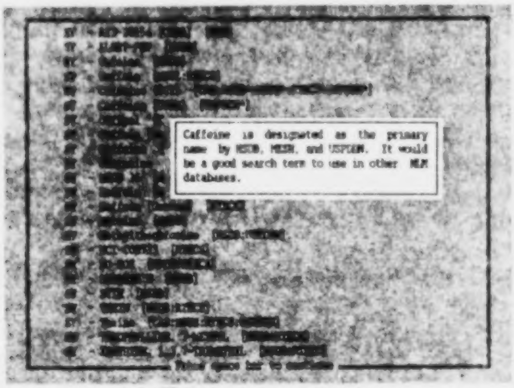
CHEMLEARN provides user-specific feedback for correct and incorrect answers to both multiple choice and fill-in-the-blank problems. The program responses to correct answers reinforce or summarize the concepts essential to the problem. Program responses to incorrect answers provide a brief explanation of the detected errors. For fill-in-the-blank problems, anticipated errors have been ordered according to their probability of occurrence. The program analyzes incorrect fill-in-the-blank answers for the most probable errors and provides an appropriate feedback. The user is given unlimited attempts to correctly answer a problem, as well as the option of having the program automatically enter the answer. A status bar in the upper right-hand corner of the screen indicates that an ANSWER is available and can be displayed by pressing the F3 Key if an incorrect answer is entered by the user.

The user has the option of accessing additional information at selected points in the CHEMLEARN program. For example, in the simulated search modules, optional HELP windows are available. These windows provide assistance to users in formulating search statements and commands. A status bar in the upper right-hand corner of the screen indicates when a HELP window is available and that it can be displayed by pressing the F6 Key. In addition, there are information windows which explain selected data in the CHEMLINE records used in the simulated searches. In the modules containing data element discussions, optional SUMMARY windows are available. They summarize and reinforce the material presented in the preceding discussion. A status bar in the upper right-hand corner of the screen indicates when a SUMMARY window is available and that it can be displayed by pressing the F5 Key. Figures III, IV, and V illustrate HELP, information, and SUMMARY windows, respectively.



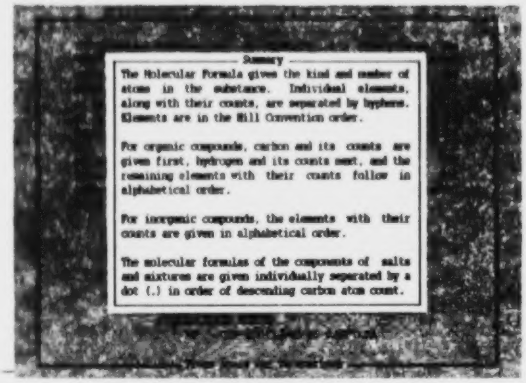
Common names are found in the SY and N1 fields. To retrieve a record using a common name, enter the first 39 characters of the name, including spaces and special characters. It is not necessary to use a field mnemonic or qualifier when searching either the SY and N1 fields.

Figure III HELP Window



Caffeine is designated as the primary name by HSDB, MESH, and USPDI. It would be a good search term to use in other NLM databases.

Figure IV Information Window



The Molecular Formula gives the kind and number of atoms in the substance. Individual elements, along with their counts, are separated by hyphens. Elements are in the Hill Convention order.

For organic compounds, carbon and its counts are given first, hydrogen and its counts next, and the remaining elements with their counts follow in alphabetical order.

For inorganic compounds, the elements with their counts are given in alphabetical order.

The molecular formulas of the components of salts and mixtures are given individually seperated by a dot (.) in order of descending carbon atom count.

Figure V SUMMARY Window

Function keys are utilized to facilitate movement throughout CHEMLEARN. Function Keys F1, F2, F3, and F4 allow the user to go directly to the main menu, return to the previous menu, advance to the next screen, and return to the previous screen, respectively. The F10 Key allows the user to exit the CHEMLEARN program and return to PC DOS.

The CHEMLEARN Map serves as an index to the program, allows the user to see which sections have or have not been completed, and acts as an overall menu for CHEMLEARN. The Map consists of a sequential listing of each of the sections and subsections included in CHEMLEARN. Subsections are indented under the appropriate section name to indicate the hierarchy of CHEMLEARN. Individual entries appear highlighted when the user has completed the corresponding section or subsection of the program. The Map may be accessed from any point in the program by pressing the F8 Key. Users may access any section or subsection in CHEMLEARN from the Map or return to the point in the program from which the Map was entered.

The CHEMLEARN Dictionary provides definitions for data elements, basic chemical terms, standard conventions, acronyms, and publications found in CHEMLEARN. It can be accessed from any point in the program by pressing the F7 Key.

A template defining the function key operations accompanies CHEMLEARN. In addition, a window illustrating the function keys and their associated operations can be accessed from any point in the program by pressing the F9 Key. See Figure VI.

The CHEMLEARN tracking function allows users to monitor their progress throughout the program. As each section of CHEMLEARN is completed, its corresponding Menu and Map entries are highlighted. Records of completed sections are maintained in a users file. This enables CHEMLEARN to keep track of the users' progress for subsequent sessions.

The CHEMLEARN users file contains the five most recent distinct user names. At the beginning of the program, CHEMLEARN asks the user to enter his or her name. If the name is currently in the users file, CHEMLEARN checks its internal records to determine which sections of the program were completed previously and highlights the corresponding Menu and Map entries. CHEMLEARN then gives the user the option of returning to the section from which he or she exited during the previous session. If the name is not currently in the users file, it is added to the beginning of the file. At the end of the session, records are written to the users file indicating which sections of the program were completed, as well as the last section accessed.

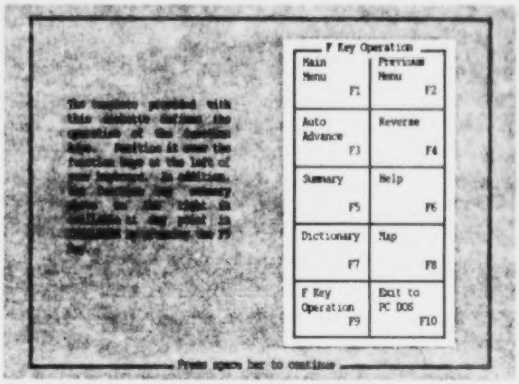


Figure VI F-Key Operation

IMPLEMENTATION

CHEMLEARN was programmed using the PILOTplus Authoring Language developed by Online Computer Systems, Inc. in 1982 (3). PILOTplus is a high level authoring language designed to assist the professional educator in developing interactive, microcomputer courseware. PILOTplus is based on PILOT—Programmed Inquiry Learning or Teaching—which was developed by John Starkweather at the University of California at San Francisco in 1969 (4).

The PILOTplus code necessary to execute CHEMLEARN is contained in over 31 files. All of the subroutines necessary for proper execution of CHEMLEARN are contained in a single file which remains resident in memory throughout program execution. As the user proceeds through the program by selecting menu options and accessing sections, each of the executable files necessary to run that portion of CHEMLEARN is dynamically loaded into memory and executed. Those files which are no longer necessary for execution of the section of the program being accessed are removed from resident memory.

The files necessary for execution of CHEMLEARN reside on two (2) low-density floppy diskettes. They require a total of 587,000 bytes of memory. CHEMLEARN may be loaded onto a hard disk prior to program execution, or may be executed directly from the floppy diskettes. If the user is running CHEMLEARN from floppy diskettes, the program prompts the individual when it is necessary to switch diskettes.

CHEMLEARN runs on the IBM-PC, PC-XT, PC-AT, and fully compatible computers. It requires 256K RAM memory, one disk drive, and DOS 2.0 or higher. No extra boards or external peripherals are necessary. It supports most monochrome, color-graphics, and enhanced color-graphics monitors.

SUMMARY

CHEMLEARN is the first in a series of PC-based, self-instructional tools for selected NLM databases. It provides inexpensive, easily accessible instruction in the use of CHEMLINE for medical and health professionals, as well as librarians and information scientists. CHEMLEARN can serve as a model for microcomputer-based training packages for computerized and non-computerized information in a wide variety of subject areas.

ACKNOWLEDGEMENTS

The authors wish to thank Ms. Cheryl White for her tireless editing, reviewing, and testing throughout the development of CHEMLEARN. The authors also wish to thank Mr. Herbert Friedman and Ms. Dorothy Moore for their assistance in the development of the CHEMLEARN Dictionary and Dr. Craig Locatis for his design assistance.

REFERENCES

(1) Schultheisz, R.J., Walker, D.F., and Kannan, K.L. 1978. Design and implementation of an online chemical dictionary (CHEMLINE). J. Am. Soc. Inf. Sci. 29:173-39.

(2) Spann, M.L. 1979. *CHEMLEARN*: A Computer-Assisted Instruction (CAI) Program for Online Chemical Information Searching. Order no. 80-8347. Dissertation Abstracts.

(3) Online Computer Systems, Inc. 1982. Unpublished work.

(4) Starkweather, J.A. 1969. "A common language for a variety of conversational programming needs." In Readings in Computer-Assisted Instruction. Edited by H.A. Wilson and R.C. Atkinson. New York: John Wiley & Sons.

WHAT'S YOUR CAI MO - MAINFRAME OR MICRO?

BARBARA HERSHEY HANDLER
OAK RIDGE NATIONAL LABORATORY

Abstract

When determining the compatibilities and differences between the use of the mainframe and microcomputer for use in computer aided instruction (CAI), each should be evaluated on its relative merits for the specific job to be done; authoring, presentation, or both. The alternatives must be evaluated regarding the content, subject matter and specific requirements of the course to be taught. The real issues are managing the CAI package correctly and being sure that the selected approach is educationally sound.

1.0 Background

The main question does not seem to be be whether or not to use Computer Aided Instruction (CAI), but how to use it. The basic discussion in this paper is whether to organize the CAI training around a mainframe or a microcomputer. The real issues that should be focused on are managing the package correctly and being sure that the selected approach is educationally sound.

In considering options, the available choices are not just the micro versus the mainframe computer. There can be a combination which would allow for creation (authoring) of CAI packages on one system and presentation to the student on a completely different system. In addition, the two alternatives can be used in combination through the process of networking.

Beyond determining the capabilities and differences between the two types of systems under consideration, each should be evaluated on its relative merits for the specific job to be done: authoring, presentation, or both. Furthermore, the alternatives must be evaluated in terms of the content and specific requirements of the courses to be taught. It is also important to consider the capabilities for administering the CAI.

2.0 Hardware Considerations

Hardware considerations for CAI include availability, responsiveness, standardization, on-line storage space, and keyboard configurations.

2.1 Availability

One definite limitation in using a mainframe instead of a microcomputer for training, is loss of flexibility. If the system is 'down,' everyone is affected and neither the training nor the application packages can run. This limitation is not as big a problem when micros are used. If one micro is not functioning properly, only one user is affected, other users are not. In the case of networking, other units can be affected if the malfunctioning micro happens to be the 'host'.

2.2 Responsiveness

Some people may argue that a mainframe system is 'slower' than a microcomputer. However, if the actual application is residing in a mainframe environment, the timing may be more realistic than on a micro. It is important for training to closely resemble the actual work situation.

2.3 Standardization

In many situations such as a university, public school, or even private business and industry, there appears to be no standard configuration for the PCs which are already in use. A minimum set of standard requirements should be determined for microcomputers running CAI.

2.4 On-line Storage Space

If the microcomputer being used for presenting CAI is not dedicated to training, then those in charge of design must consider whether there is enough space to handle current and future CAI projects in residence with other application packages. This must certainly be considered in light of the increased use of concurrency and co-residency in CAI.

A stand-alone personal computer with only floppy diskettes may not be adequate for delivery of many CAI packages due to the cumbersome nature of 'floppy swapping.' A user friendly CAI should reside on a hard disk to minimize the logistical problems of handling disks and to maximize the process of learning and attuning to the subject matter. A large area of 'permanent' storage for programs and data can be made available through the use of an internal or external hard disk drive.

If a hard disk drive, with a minimum of 10 megabytes, were available on each of the PCs used for CAI, most CAI could be loaded once from floppy diskettes and used for the duration of a course with a minimum amount of disk handling. At the current time internal disks with up to 100 megabytes of storage are available. As the number of CAI modules increases with the implementation of additional CAI courses, the new courses would have to be loaded initially from floppies to a hard disk, but the core of the system could remain resident on the hard disk. An advantage of this alternative would be that users would not have to be concerned about the care and handling of diskettes. Many users are anxious about using computers and they do not need the additional anxiety of worrying about being responsible for the care and preservation of 'destructible' diskettes. There would be an initial period at the beginning of each new course when a student, or administrator, would be required to handle a series of floppies. This manipulation would be minimized during the remainder of the course.

One foreseeable need for a floppy diskette would be to provide student lesson status information to the instructor/supervisor. Many existing CAI programs have some form of internal record keeping. Depending on administrative policy and procedures trainees may not have to bother with diskettes to report lesson status. The supervisors could be responsible for accessing such data from the microcomputer. If records are maintained locally on a single PC, individual students must always use the same PC in order for their records to remain current. For people considering authoring CAI programs for microcomputers, there are many authoring systems which include some form of internal bookkeeping and student status file creation. The use of such an authoring system would eliminate the need for the user to handle a lesson status diskette.

2.5 Keyboard

If the same keyboard were used for training and application purposes, users would not have to be concerned with a transfer of learning. Therefore, if the training is done on the same type of keyboard used at the workstation, the student has an opportunity for more realistic practice and an improved rate of learning. When a student is ready to apply the knowledge gained from the CAI, there should not have to be any 'unlearning' taking place. It is best to train users on a micro if they are going to use a micro, while users who will work with applications residing on a mainframe should receive their training on a mainframe. However, if a microcomputer were being used as a terminal for a mainframe then the transfer of motor skills would not be a problem.

If PCs are used for mainframe application package training then it is necessary to consider important factors in the use of different keyboards such as key function, placement, and 'feel.' Key function is the easiest to modify and can be accomplished with software. By using custom software tools, each key on the micro keyboard can be made to function like any key on the mainframe terminal. Alteration of key placement and feel would require significant hardware modifications. A primary difference between keyboards is the placement of the function keys. Some keyboards have the function keys placed horizontally above the standard row of number keys while the function keys on other micros may be in two columns to the left of the main keyboard. The function keys are commonly used in both training and application packages.

Aside from changing the function of several keys on the microcomputer, it may be advisable to change the appearance of these keys. This could be achieved through the use of plastic key caps; however, this would pose storage and maintenance problems. This type of modification would still give the student a different 'feel' than using a keyboard with the same key placement and action.

Another alternative to consider might be to identify a replacement keyboard for use in training. This keyboard should be similar to the actual keyboard the user would be working with after training; it should look and act like the mainframe terminal keyboard. This approach would provide the best environment for implicit learning and would be preferred over software simulation and alteration of the microcomputer keyboard. However, if the micros were already being used for specific applications that require the use of the standard micro keyboard, the alternate keyboard would only serve the CAI function and would have to be replaced by the standard keyboard as needed. Unless dedicated microcomputers could be used, someone would have to assume additional responsibility for changing and storing these alternate keyboards. The logistics involved could present a management problem.

3.0 Software Considerations

CAI development on a mainframe could require the use of a programming language rather than an authoring language since the number of commercially available authoring packages for the mainframe is less than the number of packages for microcomputers. Requiring knowledge of a formal programming language limits the availability of people capable of designing and developing CAI.

There are many more options to consider when one is thinking about development of CAI packages to be used on a microcomputer. A variety of authoring languages and authoring systems, with extensive features, are available for most microcomputers. Use of these languages/systems requires considerable financial investment and additional training for CAI producers. It is possible that the purchased software could still lack desired features. A cost analysis would be in order before deciding to purchase any authoring package.

3.1 Portability

A current question to consider is whether a micro or a mainframe should be used for authoring and delivery. Courses developed on a mainframe could be ported to a micro for delivery. Many existing CAI programs can be easily ported to microcomputers from a mainframe environment. CAI courses are often developed on a microcomputer for presentation on a mainframe. In either case, additions and/or changes should take advantage of features which are unique to the target system.

4.0 Presentation Features

The mainframe, minicomputer, and microcomputer can be used for teaching, given the appropriate CAI package and a capable authoring team. It is generally accepted that it is easier to create a glamorous CAI package on the microcomputer due to the ease of creating lessons using sophisticated graphics and sound. On the other hand, a novice may find the array of options, features, and capabilities overwhelming. As with the creation of any teaching media, the expertise and creativity of the author are prime factors in the overall functionality of the final product. The microcomputer would be the least demanding of the authors creativity since graphics could be designed with less difficulty than on the mainframe.

During a general discussion at a recent Computer Based Training Conference, a training director who heads a CAI training program for a large insurance company indicated that trainees quickly begin to find the 'jazzy stuff' annoying and, sometimes, insulting to their intelligence. The insurance company's training program makes use of both micro and mainframe CAI delivery. The director does not deny the need for a stimulating presentation so as not to bore the learners, but mentions that it is important not to impede the learning process either!

All the cosmetics a developer can apply to CAI will not make up for poor instructional design. Instructional design is of the utmost importance for all types of hardware under consideration. On the other hand, even the best design can be ruined by the misuse of special features such as color, graphics, animation, and sound.

4.1 Graphics

The graphics on most mainframe CAI systems are limited to the ASCII characters and a small set of graphic characters. However, these characters are sufficient for the graphic needs of many application packages. If it is only necessary to reproduce, simulate, or use actual application screens, then the ASCII characters are usually adequate. If the use of more 'interesting' graphics is deemed necessary to the learning process, then the microprocessor may be a better machine to use for training purposes.

The subject matter being taught is of prime importance in determining the features needed in the CAI. The requirements for CAI usually include teaching computer applications. Many application packages do not include the use of graphics. The CAI package, using simulation of application screens, may need to contain nothing more graphic than straight line drawings in which case, the mainframe may be fully adequate for the presentation of application instruction.

4.2 Animation

Animation is an extension of the use of graphics. While animation is usually not found in an application package, it can be used successfully to present material to the learner. It is important that the learner remember the material and concepts presented and not just the 'cute' presentation. Animation can be used to represent such essential things as the data and information flows within the application package.

4.3 Color

The argument for color is similar to the argument for graphics. If color is not necessary for the application package it, therefore, is not 'necessary' for the training package. There is no doubt that the training may be more interesting with the use of color, but color is not an essential. The overall training package may be enhanced by the use of color as an organizational tool. Colors may be used to represent various menus and templates allowing the user to develop visual cues regarding such things as his/her place within the overall training and the success or failure of responses to questions or exercises.

4.4 Sound

Sound is used in a limited manner in some CAI and few application packages. Many systems indicate an error by 'beeping'. Most users don't mind this simple reminder since they might overlook the error otherwise. Both the mainframes and the microcomputers are capable of producing the beep sound. Any other sounds seem irrelevant in computer application training.

5.0 Communications

If several PCs were networked together, it would be possible to let other PCs on the network access the "host" microcomputer where the CAI system is stored. This option would allow centralized monitoring by a human administrator. If the network does not already exist, installation could be an obstacle. The ease and cost of networking PCs would be dependent on the number of PCs to be used for CAI, their proximity and their degree of similarity. Networking would lend itself to supporting the use of centralized data files which simplify administrative duties by simplifying access to stored data. It should be mentioned here that with a mainframe, the network is part of the design.

The 'best' alternative may be a combination of local storage methods on each PC, and networking. Such a system might use local media for both screen information storage and generation of the screen locally while using the network and central files for storage, modification, and retrieval of shared data. This shared data would be in the form of session logs, and broadcast messages. Such a system would provide the greatest flexibility and ease-of-use for the students, administrators, and courseware developers.

6.0 Course Administration

The course administrator is responsible for course maintenance and handling the reporting capabilities within the CAI.

6.1 Course Maintenance

An essential point in favor of the central computer is the ease of making changes to a training package. Once a change is made, it becomes an integral part of the program. When the computer being used for training is a micro, changes can be made easily although duplication and distribution of updated courseware may present problems. The cost of an update to a micro system is probably greater than the cost of an update to a mainframe system. The distribution of updated diskettes could require a sophisticated bookkeeping system. It is possible that 'old' versions would remain available and could even be used interchangeably with the new versions, thus causing confusion on the part of the student and the administrator. The above argument is not meant to imply that updates to a central computer are trouble free. The problem of implementing changes at different mainframe sites is very real. However, the number of sites is usually less than the number of individual terminals or PCs, thus, there is less to manage and less time is needed to complete the updating process.

6.2 Reporting Capabilities

Often the reports made available on a mainframe CAI can not be produced in a micro environment without a significant increase in the amount of work being done by an administrator. It would become an additional responsibility for a supervisor, or other designated person, to gather the information from each individual's disk and store it in a central location where it could then be used to produce composite reports.

Unless the PCs used for CAI are networked together, the CAI instructional modules and the CAI administrative modules might have to run on isolated PCs. Without network communications, an alternate method of transferring information is required. Data that should be accumulated includes such items as student progress, test results, and student evaluations from each student's PC to the PC being operated by the supervisor. Without a network, this can best be accomplished by sharing floppy diskettes. The multi-user capability of a central computer eliminates the need to collect physical media from each student.

The use of non-networked PCs presents a distribution method that complicates and detracts from the independence of supervisor-student interaction as provided in mainframe CAI packages. The supervisor would not have access to information about a student until that student submits a diskette containing the needed information. The students using a microcomputer do not have the benefit of receiving broadcast messages or be able to see general announcements and update information that may be generated in response to questions asked by other students who are enrolled in the course. If students were required to get a floppy diskette from the administrator at the beginning of each CAI session and turn it in at the end of the session, they would lose a lot of the flexibility of setting their own schedule and moving through the course at their own pace. The time required of the student and the supervisor would be increased if the CAI were presented on individual PCs, not to mention the possible damage to or lose of diskettes.

An additional concern is that handling of diskettes, by students, makes the information on the diskette less secure. What would prevent another student from using a diskette that didn't belong to him/her? This may not be a major problem but it could counteract any measures taken to provide security for the CAI in general.

When considering the use of diskettes, one must be concerned with the problem of backup. It is essential that floppy diskettes be duplicated to protect the information on the disk. Thus, if a disk 'crashes' or is otherwise damaged, the majority of the data is not lost. Lost and/or damaged diskettes present problems in the areas of additional costs, physical storage, and security.

7.0 Summary

The information presented in this paper represents a general survey of the advantages and disadvantages of using mainframes and/or microcomputers for the authoring, presentation, and administration of computer aided instruction. It is not an exhaustive study but, rather, the presentation of ideas to consider when making decisions concerning the designing, authoring, presentation, and administering of computer aided instruction. It is recommended that the designers of any computer based training consider hardware, on-line storage space, communication problems, keyboard configurations, application system structure, the USER, importance of graphics, course administration, available authoring systems/languages, principles of education, and the ultimate goal of training before deciding on the authoring and delivery method for any CAI.

Dr. Barbara H. Handler
Martin Marietta Energy Systems, Inc.
P.O. Box X
Building 4500N, MS 207
Oak Ridge, Tennessee 37831-6207
(615) 574-0262

TRAINING VIDEOTAPE PRODUCTION AS A MEANS OF STUDENT CONSULTANT TRAINING

JOAN LAUER HAYES

WESTERN WASHINGTON UNIVERSITY

ABSTRACT

A 17-1/2 minute videotape was produced to show the importance of "contracting" (whereby student consultants can maintain control in even difficult interactions with users), and body language, in consulting computer/lab users The actual production, including script writing, became a valuable training project in itself. The tape will be a part of the presentation.

INTRODUCTION

For some years as the supervisor of a student consulting staff now numbering more than 60, the author has been frustrated in achieving an important goal: How does one make it possible for staff to truly be aware of their consulting styles and the effect on users?

The most successful efforts included games and role plays, which were valuable for the participants but short-term in their effect. Since many of the student staff come to us inexperienced in either computing or providing services, we were looking for a mechanism which could depict clearly and uniformly the points we were trying to make. It occurred to us that a videotape with opportunities for discussion would provide training which would be relatively uniform in each of the nine computer labs on the Western Washington University campus.

It was easy for us to recognize that we were starting with complete inexperience in videotape production; we decided to assume that we also knew nothing about good consulting. The latter was a difficult and important posture for the staff members who felt the most expert and least drawn to self-criticism; their willingness to rethink their roles was one of the most valuable outcomes of the project.

The easiest place for staff to focus on the consulting process turned out to be things they did not like or which were difficult: we generated lists of unpleasant user comments and stressful situations. (Figures 1 and 2.) We also generated lists of positive user characteristics. After formulating items which typified "good" and "bad" consulting, we abandoned the latter. It had become clear that we would be focusing on difficult users and good ways of dealing with them, and that to talk about bad consulting as if it were ever-present had a demoralizing effect on the staff.

Since we were beginning to get a picture of what might produce effective consulting exchanges, we developed a code for recording observations of each other. (Figure 3.) Each consultant made a written observation of at least one other person on duty. Where possible, user behaviors were noted as well, but most often observers were kept busy recording only the consultant behavior.

We still had not decided on the focus of the videotape! After probably four hours of discussion in as many weeks of meetings for each lab, there was consensus that most problems lie with users and not consultants, and that probably there is not much that we can do to make major changes in users. But ideally, any consultant should be able to deal with any user. After recognizing that staff members, also, come in a variety of temperments, the decision was made!

We would show three consultants, of three different personality types, working with the same mildly unpleasant user. Our goal was to point out to new staff members that no matter what their background or inclination, they will be able to consult effectively. The three types, Aggressive, Timid, and Overeager made it possible to incorporate many of the problems with consulting which were elicited earlier in the process.

Here we were, with the user and staff members pretty well defined, but no way to keep the user from running roughshod over each of the consultants. Pam McRae, the WWU Tutorial Center Coordinator, came to the rescue with a technique which has worked well for her staff. Labelled "contracting", it becomes possible for the consultant to maintain control of the exchange by stating what it is possible to do and getting agreement from the user. So we spent several more lab meetings talking about contracting in the lab setting.

At last, we were ready to begin script writing! Staff selected their "favorite" personality type and worked on different teams. (One of the most important was the cadry of word "processors" who kept up with the many revisions generated by script writers and editors.) Writers were encouraged to work from existing lists of verbal and body language behaviors. And they were also encouraged to keep the events realistic even at the expense of some humor.

The Media Services staff who provided technical support for our project were also willing to spend many hours providing information and guidance. At their suggestion we hired a fully-qualified director and a drama coach, both of whom were interested enough in the project to charge small fees. (Figure 4.) Based on previous experience Media Services personnel also encouraged us to select trained actors; we stuck with the original plan to keep it "our" production by using student staff members.

We were able to tape in one 13-hour "day" what the taping crew said would ordinarily have taken two to three days. For that, we credit the experience and cohesiveness of the shooting crew, the skill of the drama coach, the enthusiasm and responsible approach by all students involved in the project, and a consistentaly organized structure throughout. At times we wondered whether all of the preparation was necessary. It did pay off. Further, the value of the discussion and preparation enumerated is clear to us all; not only would the quality of the videotape have suffered without it; we would also have been deprived of one of the most productive training exercises in the history of WWU Computer Center lab staffing. The quality of consulting has improved markedly this year, and the recently completed videotape has yet to be used.

FIGURE 1

Questionnaire re Characteristics of Computer Users For Videotape Use

In order for the videotape scenarios to be realistic and reflect typical exchanges between consultants and users, we need specific data. This data is hard to collect because it involves recognizing subconscious actions and natural movements. Now we need you to think objectively and thoroughly about those.

1. Please list characteristics of an unpleasant user:

tone of voice ______

actual words ______

posture, hand position ______

signs of nervousness, body tension ______

eye contact, facial expression ______

other body language ______

response to your suggestions ______

response if you are busy ______

other ______

______

The user treats me like a: ______

2. Please list the characteristics of an effective consultant:

tone of voice ______

actual words ______

posture, hand position ______

signs of nervousness, body tension ______

eye contact, facial expression ______

other body language ______

response if you are busy ______

response if you don't know the answer ______

other ______

______

When I am an effective consultant, the user treats me like a:

______

FIGURE 2

Characteristics of an Unpleasant User

Tone of voice

Angry, unfriendly, demanding tone.
Hurried, mumbled, spoken while not facing consultant.
Loud, fast.
Whiny, loud, so soft it's hard to hear, or sigh.
Very loud, abusive.
Authoritative (show me, tell me), whining demanding.
Loud, grating.
Irritating, whiny, frustrated.
A whiner (like on Saturday Night Live).
Nervous, hesitant.
Impatient, not trying.

Actual words

"I need some help over here", or "I should be able to use WordStar."
"How do you save a file?" over and over and over.
Rude, vague, show no forethought.
"I just want to know how to ...", no comments, interrupts.
Self-demeaning or invalidating.
"I'm in a hurry."
"I don't see why you don't know." "Do this for me."
"That's not what I asked for."
"I can't see why this won't work."
Commands "Come here", or "Fix this".
Swearing at computer, blaming everything but self.
"How come this won't work?" "What's wrong with this thing?"
"I just want you to show me."
"I don't understand. What is happening?"
"How do you do this?" "I don't understand." "What is it doing?"
"I don't know how he expects us to do this."

Posture, hand position

Leaning forward, shoulders back, eyes bugging out.
Hunched over.
Relinquishing control.
Tense, hands flailing; on hips, leaned away.
Stiff, holding papers, waving papers.
Upright, threatening stance, hands out.
One hand on hip, the other on the desk.
Defensive (arms crossed, etc.).
Violent arm movement.
Slouched, slumped.
Eyes glued to screen.
Not typing -- wanting you to enter it.

Signs of nervousness

Hesitant hands.
Frustrated, anxious.
Tapping, rocking, gripping pencils.
They look disgusted with situation.
Nervous, excited.
Massive uncertainty, stuttering, etc..
Tapping foot/finger.
Fidgeting.
Tense, uptight.
Pushing hair away from face.
Sweating.

Eye contact, facial expression

Direct contact, without smile or a smirk.
Confusion, eyes on screen.
Not focused on problem, worried.
Eyes large (scared), not looking at me, no smile.
Little eye contact.
Staring, grim look.
No smile; frown.
Not paying attention; i.e., waiting for the answer.
Angry, glaring.
Dejected looking.
Clenching teeth, staring at other computers.
Dazed.
Little eye contact; stares at the screen.

Other body language

Turning away when talking to them.
"You fix it" attitude.
Self-demeaning, or aggressive.
Rocking (nervous but coping), jaw clenching, hands on hips, tense facial muscles, nail tapper, foot tapper.
Face very tense, eyebrows tightened.
Slam down, swear.
Foot tapping, shifting weight.
Huffing, puffing, sighs.
Demanding.
Slumped in chair.

Response to your suggestions

Either no response or maybe a smirk or grunt.
"Well, I tried that already."
Expects it to produce perfect results immediately.
"That won't work."
Try things quickly -- making mistakes as they go along making matters worse.
"It won't work."
"No, I don't think that's the problem".
Argumentative, "But what if ...", Doesn't believe you.
Ignoring it.
"I tried that already!!!"
Forgets it and asks again immediately.
"Wait, let's do this a few more times."
"But I don't know what I am doing."
"I don't/can't do that", "I don't know how to do that."
"Show me how."

Response if you are busy

Not understanding, "I really need help now."
Heavy sigh.
Sigh or curse, angry look.
Thinks I should be there immediately.
To leave, or to complain.
Impatient, wants help now.
Shakes head and walks away.
Not concerned whether I'm busy or not.

Will demand as much time as they want.
Impatient, walks away.
Grumble, stomps off.
Impatient, pushy.
"I can't wait", "Help me NOW!"
"I need help over HERE." "Can you help me?"
"Hey! what about me?"

Response if you don't' know the answer

Unappreciative for the help given.
mumbles "Those consultants don't know anything."
Look of disbelief, "Well, how will I find out?" Wants to put responsibility on me.
"I should have known it was a mistake to ask."
"Well, does someone else know?" or "I'll just have to wait to ask the prof."
"Forget it."
Will not be satisfied with an "I don't know."
Expecting an answer.
"Some consultant you are."
"Hold on for a second."
"Well, what am I supposed to do?"
Irritated.
"Does anyone else know?"

Other

Expecting the consultant to give an immediate answer and do pieces of the work for the user.
Assumes I have the authority to relax any or all the rules of the Computer Center.
"Hot shot" attitude.
The user sometimes becomes very bossy, apparently to give the impression he/she knows more than everyone else.
The user has used the wrong terminology to explain what the problem is, therefore, I'm not answering the right question.
Person is in a hurry. User won't take the time to look it up. Demanding, not polite.
Fear of losing grade points.

"Users treat me like":

I should be glad to be helping them.
A Saviour.
A "quick answer" machine, or one who single-handedly developed the IBM-PC's or the VAX.
I should know everything.
A lower form of life.
They are totally right and I'm wrong.
I feel like the users want me to do the assignment for them.
"Know all."
Monarch, usually with respect.
Person who should know everything and when I don't, a "Then how did you get a job here?!"
Dictionary, encyclopedia.

FIGURE 2A

Characteristics of an Effective Consultant

Tone of voice

Friendly, pleasant, touch of excitement.
Calm, reassuring if necessary.
Relaxed, controlled, medium range.
Authoritative, smooth, low tone.
Firm, clear.
Always perky, eager to help.
Calm, rational.
Pleasant, inquiring.

Actual words

"Sure, I'd be happy to help."
"How do you think we should ...?"
Polite, demonstrating real understanding.
"Let's take a look," "What exactly were you doing?"
Encouraging, responding to user's expressed or unexpressed needs.
Sound reassuring. Repeat what the user has said to make it clear that you have heard.
"Can I help you?"
"I understand; and here is how to fix your problem."
"Do you understand?", "Anything else?"
"Hi! can I help you? Is everything O.K.?"
"Hi, can I help you?"
"What do you need?", "How can I help you?"
"Let's try to figure this out", "What is the trouble?"
"O.K., let me summarize your problem and see if I can help you out."
"Good job!"

Posture, hand position

Relaxed posture, hands at front or side.
Leaning toward user.
Open but not too active; i.e., more directions than actions.
Relaxed, leaning forward.
Stand where appropriate, lean forward; if seated, eyes up and face alert.
Point to what you are talking about.
Open posture.
Explain graphically (drawings, etc.).
Interested.
Leaning forward.
Kneel next to machine, hands off keyboard.
Usually hand is pointing to the screen.
Hands down, open posture.
Relaxed, kneeling (at times).
Stand beside.

Signs of nervousness, body tension

Preferably relaxed, not nervous.
Just enough tension to show awareness.
Lack of fidgeting.

Eye contact, facial expression

Some eye contact without staring; smiling pleasantly.
Frequent eye contact, relaxed face, smile.

TRAINING VIDEOTAPE FOR STUDENT CONSULTANT TRAINING

Look in their eyes and smile or look interested.
Good eye contact, concerned look.
Good eye contact but doesn't "stare".
Eye contact is necessary so users feel you are talking to them. Smile.
Smile, look in the eye.
Happy face, occasional winks at girls.

Other body language

Sitting in chair next to user, is possible.
Stands back allowing user to act.
Eagerness, but not too eager.
Gesture with hands - body.

Response if you are busy

"I'm really busy at the moment -- I'll help as soon as I can."
"Go thru[sic] menus; I'll be back."
Recognize user's presence.
"I'll be with you as soon as I'm free."
"I'll see you and will ...", "Let me see if I can find __________ to help."
Explain that you are busy and that you will be with them in a moment. Let others who are waiting know that you will be with them also.
"I'll be with you in a minute."
Quick, to-the-point answer.
Why don't you try or look there ... .
"I'll be right with you."
"I'll get there as soon as I can."
"Hold on, I'll be there in two minutes."
"I will help you as soon as possible."
Acknowledge them and tell them you will be there soon.
"Please wait a moment until I finish what I am doing."
"I'll be there as soon as I'm done here."

Response if you don't know the answer

Suggestions where to go or look for an answer.
"Let me go get someone who does know."
Suggest either a source for the answer or some means of figuring it out.
"Let me check with the other consultant," or "Let's check the manual."
"I'll find out by tomorrow." "__________ can help you at _____ pm."
If you don't know an answer direct them to a manual, say you don't know, direct them to someone who may know, i.e., a prof., a consultant, another user.
"Sorry I can't help you but ________ will be in soon and he/she can help you."
Is there a good response?
"I'm sorry, I don't know, ask Tim!"
"I don't know, ask your instructor."
"I don't know, try _______."
"Steve, we can't seem to figure this out. Do you think you can give us a clue to what's wrong here?"
Look it up or refer to prof.

"When I am an effective consultant, the user treats me":

Thankfully.
Like a human being.
Like a reliable source of information concerning general questions.
Like a peer with a different role, not as a superior.
Like a new friend.
Like someone who has helped them out a lot and makes them very appreciative.
A reference or referral person. "Human" consultant.
A friend.
A valuable resource.
A friend who can help them.
Important learning tool.
A helper. A friend.
Helpful resource.
Professional and intelligent person.
A valuable person; someone who actually knows something.

FIGURE 3

Observation Code for Videotape I

U	User
C	Consultant
D -->	Right of Desk
<-- D	Left of Desk
D	Front of Desk
ST	Both Standing
SI	Both Sitting
CS/US	Consultant Standing/ User Sitting
US/CS	User Standing/Consultant Sitting
U3C	3 feet between user and consultant
U5C	5 feet between user and consultant
U5+C	Over 5 feet between U and C
LF	Leaning forward
LB	Leaning back
M	Arm motions
IC	Eye contact
Ftap	Finger tapping
FTtap	Foot tapping
T	Tense
R	Relaxed
H	In a hurry
f	Loud voice
p	Soft voice
‿	Smile
⌒	Frown
—	Inbetweener
TY	"Thank you"

FIGURE 4

Western Washington University

Computer Center Training Tape

Budget

-	15 hrs. Meetings & rehearsals x 5 students = 10 x 15 x $4.00 =	$ 300.00
-	Script writing: 3 scenes x 2 students x 2 hrs. x $4.00 =	50.00
-	Taping: (Cast salaries) 4 students x 6 hrs. x $4.00 =	100.00
-	Editting: (Value $150)	NC--
-	Taping: (Value $360)	NC--
-	Incidentals:	20.00
		$ 470.00
-	Director, 12 hrs.	400.00
-	Drama Coach	100.00
-	Narrator	25.00
-	5-3/4" Tapes	75.00
	Total	$1,070.00

Director's

12 hrs. Include:

2 hrs. Introduction
6 hrs. Script
2 hrs. Misc. shots (T shots)
1+ hr. Setup, etc.
1+ hr. Breakdown
1 hr. Meeting with cameraman

Joan Lauer Hayes
Computer Center
Western Washington University
Bellingham, WA 98225
206-676-3367

THE IMPACT OF TRAINING IN ENLIGHTENED COUNTRIES

John J. Hirschbuhl

The University of Akron

Abstract

Although training has not attracted the attention it deserves, some countries are increasing their commitment and beginning to reap the benefits. In the U.S. a number of organizations have made appreciable gains using technology for training, but there is still no national consensus of quantifiable results. In Asia, however, where society has leaped suddenly into the technological age, there is both social and political impetus for taking full advantage of current training delivery systems. What lessons can be learned from these experiences? How can corporate and national attitudes be changed? How soon will we lose our competitive edge if we do not make full use of hi-tech training systems?

Learning from our Industrial Past

Throughout the last century enlightened countries in Europe and the U.S. have looked to breakthroughs in machinery and systems design as the principle means for realizing gains in productivity. Unfortunately, little attention has been paid to making the work force more proficient through training. History has recorded the rise of "sweat shops," which lead to national trade unions that resisted implementation of automated productivity tools. This contributed to the ensuing loss of our competitive edge to Asian nations and resulted in the near demise of industrialized society in Europe and the U.S. The mistakes of the past seem to have been recognized in the U.S. during the 1980s. The search for ways to develop and nurture a more highly motivated work force has been a paramount concern of American industry during this decade. There have been attempts to achieve greater participation in management, and there is wide-spread implementation of training technology in the U.S. American business leaders have realized that competence in individual job roles is the bedrock upon which to build a motivated workforce. Therein lies the basis of their quest for sound, effective corporate training programs that are delivered by an array of instructional delivery systems. These delivery systems include trainers and current training technology.

The Rise of Computer Based Training in Corporate America

The first magic moments of the Computer Based Training (CBT) build-up began in 1981 when the price of the average American automobile passed that of a generally usable business computer. During the five years that have passed, the price differential has grown wider and corporate America has been inhaling personal computers at a rate of about $25 billion. Because of this ingestion, CBT is being woven into the greater web of corporate training. In 1985, International Data Corporation conducted a survey to determine the level of CBT usage within major American corporations. The results of the survey indicate that 66% of those who responded were already using CBT in one or more training areas (technical skills, management/sales, administrative, clerical, programming, computer concepts, and training management). CBT has become one of the primary training delivery systems within such companies as Eastman Kodak, Exxon, Ford Motor Company, General Electric, General Motors, Hughes, IBM, McDonell Douglas, Rockwell International, and Xerox.

The Quest for Quantifiable Results

Although the population of CBT installations and applications in the U.S. is experiencing phenomenal growth, there is a paucity of data that gives any clear indication of CBT's effectiveness in corporate America. Part of the problem may be corporate America's unwillingness to make public the results of its training programs. However, the problem may also be due to an assumed level of adequacy and lack of motivation to investigate the effectiveness of CBT once the business case has been made. There appears to be an unconscious acceptance of CBT's adequacy and a lack of desire to reach maximum effectiveness. Unfortunately, many of the training programs fail to meet the standards of adequacy necessary for maximum effectiveness.

What are the Standards . . . The Need for Learning Management Systems

Many CBT programs fail to incorporate the attributes of being able to assess and monitor learner progress and make the accruing information available in a form that can be acted upon in "real time." Collectively, such attributes fall into a category referred to as "learning management" which includes the capability to perform the following functions:

- Achievement testing--in all of its forms (placement, diagnostic, mastery)
- Achievement record-keeping
- Achievement/status reporting
- Assignment of appropriate learning prescriptions.

Without such a learning management system, one can neither be accountable for what has been achieved through the training nor optimize the CBT programs to achieve maximal effectiveness. Corporate America needs to re-evaluate the benefits of reaching for excellence. It was this same outlook in manufacturing that permitted Asian nations to overtake the U.S. in the mainstay industries of steel, automobiles and electronics. The U.S. failed to implement quality control (production management) into its manufacturing systems and, hence, failed to make them maximally effective and efficient. In addition to quality control, design and production techniques that provide the right components and structure within the courseware must be applied.

What Are the Components of Good Computer Based Education Systems?

To train people effectively by using a computer you should provide the following:

1. Interactive graphics.
2. Individual instruction.
3. Positive feedback.
4. A high degree of interaction.
5. A compact and succinct student text.
6. A comprehensive instructor's guide.

The Structure of Effective CBE Courseware

In order to build CBE courseware that can be highly efficient and effective, there are certain elements that it must contain. The following is a suggested array of requirements that the courseware should contain:

1. Specifies the target audience.
2. Manages the learner's expectations "up-front."
3. Specifies the required entry competence of the learner.
4. Outlines the range and scope of the content.
5. Specifies the overview, objectives and key ideas.
6. States the objectives in terms of the learner's desired achievement.
7. Includes higher order objectives.
8. Uses pre-instructional activities such as:
   - o Placement tests
   - o Advanced organizers
   - o Titles at the beginning of each unit or activity.
9. Formats the screen for maximum communication; that is, the upper two-thirds of screen are for graphics and the lower third is for text and response.
10. Reinforces graphics with text.
11. Uses graphics appropriately.
12. Provides interactive graphics for tutorials, practice and simulations.

13. Provides corrective feedback when incorrect answers are given by the student during practice or simulation exercises.

14. Provides for student inquiry through hints and interactive dialogue.

15. Provides real-time prescriptions based on student performance.

16. Provides "how-to-use" instructions for the student.

17. Provides a manager's guide with answers to all exercises and mastery tests.

18. Uses color effectively for background and graphics.

19. Provides evaluation components in the form of on-line pre- and post-assessments.

20. Uses a choice of behavior model for implementing the Computer Based Education activity.

The Features of the System

Systems that deliver instruction in one delivery mode are less effective than those that have multiple delivery modes. Systems that can only support text delivery are not as powerful or as effective as those with a variety of modes. For example, those systems that have graphics, sound and color can deliver much more effective instructions than those that are limited to monochrome text.

We now have low-cost systems that can provide interactive color graphics, text and sound through a stand-alone or networked environment. One of the important things to look for in such systems is the way in which graphics are used to deliver interactive instruction. Systems that deliver full-screen graphics but do not take advantage of the build effect that icons can provide do not take full advantage of the computer's interactive capability. For example, such things as interactive coding sheets and flowcharts mixed with pseudocode can provide very powerful instruction. Full-screen "pretty" pictures do not have enough instructional power to build complex concepts.

Another important component of a good CBE package is a compact and succinct text. This should provide an overview of each course, its objectives, key ideas, test items and associated exercises, such as programs to be written and executed or systems to be designed. The text can serve as an advanced organizer or a portable review facility.

In addition to these features, good courseware provides plenty of interaction and profound feedback. It is the combination of these two features that give CBE its instructional power. Responses such as "terrific" to trivial questions are a major misuse of the CBE tool.

One of the key features that CBE provides is its data collecting and management facilities which enable education managers to handle large numbers of students and be accountable for the education activity. Good management systems eliminate the logistic problems that go with classroom instruction. Good data collecting and reporting facilities enable trainers to account for the education on a student-by-student basis. They also make it possible to cost analyze and justify the education expenditure.

What Kind of People are Required to Build CBE Courseware?

Good CBE teams include the following:

- A poject manager/manager of quality assurance
- A designer/editor
- A developer/programmer
- A graphic designer
- A print materials specialist
- A subject matter expert.

The manager manages the project and performs quality assurance. He is not responsible for getting the courseware finished on time, but he must ensure the quality level of the finished product also.

The instructional designer designs the courseware. His responsibilities include formulating an instructional strategy, writing the course and enabling objectives, conducting task analysis, sequencing and grouping tasks and many other related activities. In addition to these responsibilities, he is responsible for editing all of the on-line and text materials including the student and instructor guides.

The developer is responsible for following the instructional design to produce flowcharts of the major modules, for writing the content of the modules and for developing graphics screens and sequences. In addition to these writing responsibilities, he must program the material and validate all of the development.

The graphic designer is responsible for screen design and all graphic development, part of which deals with the use of icons to build concepts by insert and overlay techniques. In addition, the graphic designer must determine color combinations (and be able to supply the drawing capability required to graphically represent themes and concepts) and be responsible for all key lining of the text materials, the print graphics and page layouts in the student and instructor guides.

The print specialist is responsible for all printed materials. This person only writes all the printed materials, and must key the materials to each of the courses in the series. In addition, he must coordinate all the key lining and page layout activities with the graphic designer and the print shop.

The subject matter expert is responsible for all content and will provide the following:

- Objectives
- Content
- Question/test items
- Final review of all materials.

Since there is a great deal of writing and developing required to prepare the material for use in a CBE program, frequent (weekly) contact with the designer and developer is mandatory. This also includes time for review of the material before and after it has been programmed.

What it Takes to Build a Course

The production stages required for an average computer based instructional unit presumes that some preliminary work on the needs analysis has been initiated by a product planning committee, the results of which will have indicted an instructional need that is believed to be amenable to a CBE product.

Here, an average CBE project is presumed to be one that will provide approximately fifteen hours of hands-on instructional time for the average student in the target population. Time estimates are reported in work days. The following is a guideline statement for estimating the level of commitment required to produce an average computer based education series.

STAGE	**DAYS**
1. Specification	20
2. Design	50
3. Development	150
4. Review cycle	20
5. Revisions	35
6. Evaluation	15
7. Final revisions	5
8. Materials, production and dissemination	10
TOTAL DAYS	305

In addition to providing a product planning activities schedule to build CBT courseware, we must define the type of engineering required to build effective and efficient intelligent courseware.

The Need for Cognitive Engineering

As with many matters, the key to maximal effectiveness and efficiency is at the cutting edge of new technologies. The current cutting edge of Computer Based Training rests on the problem of enabling computers to capture the knowledge and cognitive skills of human problem solvers. The cognitive net used for capturing is woven with the computer code which is used to encode the captured knowledge into the computer. Capturing the knowledge and cognitive skills of human problem solvers provides the required resource for building expert systems (smart CBT). In order to build adequate and efficient expert systems, cognitive engineers must determine the validity, reliability, specificity and internal consistency of the opinions rendered by selected expert problem solvers.

Once the expert system is built, the cognitive engineer must collect performance data on the system's ability to resolve the problems it was designed to resolve. Unfortunately, in the U.S. there is a paucity of performance data on operational expert systems. In fact, many professionals who purport to develop expert systems seem genuinely puzzled by the query, "Can I see the performance data?" Often there are no data. The "success" of a system is usually in the mind of its builder. This situation is obviously unsatisfactory from the standpoint of knowledge building applications. The fundamental question faced by the expert system designer is whether or not a system can consistently provide adequate solutions to the problems at hand. It is unfortunate that data from tests of expert systems in operational settings are almost nonexistent.

Recommendations

The success of expert systems is contingent upon finding effective solutions to several recurring problems. The knowledge and skills employed by experts must be modeled, encoded into a computer, and validated in an objective manner. The knowledge engineering process, the encoding scheme, and the performance assessment problem all deserve considerable research attention. The design of the human-system interface has many problems, most common of which is the need to explain the system's reasoning--that is, how it reached a conclusion or recommendation. There is a pressing need for a new kind of task analysis--one that characterizes the cognitive tasking of humans in a way that identifies what should be automated. For example, if a contemplated system requires humans to assess more than a few information sources, consider more than a few rules per decision, or make more than a few decisions every minute or so and, if uncertainty in the information sources or decision rules is not significant, then automation is probably indicated. There are no ready, generic answers to such questions as 'How "few"?' or 'how "uncertain"?' More research and applications are needed so that humans can build their expertise on developing and implementing expert systems. Currently, in the U.S. expert systems are being used more than they are being researched. This is both a benefit and a liability.

The Situation in Asia

As we look at Asia, we can see a growing demand for CBT and expert systems. Unlike the activity in the U.S., the Asian movement is government backed; it has national purpose. In countries such as South Korea, Thailand, Singapore and Malaysia governments want to use CBT to maintain their stability by providing mass implementation of the technology into the country's learning centers. They are attempting to quiet civil unrest by building hope within the population--hope which comes from upgrading the people's skills and, hence, giving them a chance at upward economic mobility. In addition to these objectives, the governments are looking into the knowledge industry as a new business opportunity. They want to export knowledge systems and trained personnel to implement such systems to nations wishing to build their knowledge skills.

In much the same way as in the manufacturing industries, Asians are studying the technology and measuring CBT effectiveness to make the systems maximally effective and efficient. In addition to conventional CBT, many of the emerging nations see telecommunications systems as an area for developing economic growth and want a larger spectrum for their use. Asians intend to use such systems to distribute the CBT facility, first to their own people and later to people of other countries. Japan's highest priority is to perfect and fine tune its computer and value-added network system, with Fujitsu and NEC competing against IBM and other successful small computer makers in every possible market within the U.S., developed and developing nations. One of Japan's future objectives will be to capture human thinking and knowledge on single chips (expanded expert systems), not to replace the human mind, but to assimilate and correlate vast quantities of data in minute fractions of time, releasing the human mind to focus on decision making and learning. The Japanese are looking at using technologies such as CBE and expert systems to change the pace at which obtaining certificates and degrees takes place. This will allow individuals to proceed at a much faster pace because access tools will make it easier to monitor an individual's progress. The major areas of impact are likely to be technical training and the Bachelor of Arts degree.

The Development of a Knowledge Industry at STOU

Currently, Sukhothai Thammathirat Open University (STOU) of Bangkok, Thailand, has a program that serves 400,000 students throughout the country. STOU has administered the entire program without CBT for several years, but the turnaround time on grades and the giving of examinations has been a very protracted activity. STOU is planning for a national CBT network that will provide learning centers at 83 locations throughout the country. It will use a telecommunications system to deliver CBT learning management systems, interactive tutorials and video-based instruction and as it develops courseware and instructional systems, it will first market them to its own people as part of a mass education program. Later, the instructional model for the delivery system and instructional materials will be exported to other developing nations who wish to 'ride on the back' of instructional technology into the 21st century. Not only will the Thais be upgrading their knowledge, but they will also be expanding their economy as a result of their knowledge export business.

In addition to these tangible benefits, the Thais will be gaining experience in using and developing an interactive technology that will enable them to do the following:

1. Individualize each student's education by building systems that provide instruction which is 'on-target' with an individual's profile of prior knowledge or skill.
2. Provide training in a responsive environment which encourages abundant feedback.
3. Provide an instructional delivery system that is adaptable--that is, if the instruction is not succeeding, adjustments in strategy should be made as quickly as possible--preferably in real-time.
4. Provide 'on-demand' instruction--that is, whenever the student is free to take it.
5. Provide transportable quality training programs to locations where and when they are wanted.
6. Provide training systems which can easily be improved over time and can be optimized using student performance data to reach maximum efficiency.

Once this industry is developed, the Thais will have a skilled group of knowledge workers whose expertise will be in demand around the world. Hence, they will have a second exportable resource: their people. These are the ingredients of which economic powers are made.

The Ironic Conclusion

The irony in the Thai plan is that the importers of the expertise during the last decade of the 20th century are likely to be the enlightened countries of the West--the nations that first created, used and developed the market for CBT will reverse their role from exporters to importers of knowledge systems. In part, this reversion is due to resistance to a national program of implementation in the West. For whatever reason, we choose to let events take their own course rather than develop national plans for adoption, study, improvement and distribution of a CBT industry. If the U.S. is to maintain or improve its position in world trade, it will have to take on a national commitment for the development, implementation, and improvement of knowledge systems such as CBT. We could then build a solid industry that provides knowledge systems for the world while we improve the skills and motivation of our work force.

There is reason to believe that we are making progress, but we are a long way from gaining dominance in the area of CBT. There have been major injections of CBT into corporate America, but we failed to put forth the effort required to make these systems maximally effective. In addition to our own resistance to advance the technology, we have met resistance from publishers who see new delivery systems as a threat to their existing book product lines. At times, it seems that we are prisoners of our past success. We are hesitant to give up on old products simply because they still have some profit left in them. Unfortunately, we may be losing the business in order to get the last ounce of profit out of outmoded products. What is needed is a strategic plan for managing the technology turnover.

Strategic Suggestions

Technology changes rapidly. Whatever system you choose will create future technology problems. What is needed is a phased approach. Plan on evolving the system over a three-year period and implementing one-third of the system each year. By year four, plan on replacing or upgrading the equipment and software purchased or developed during the first year. By year five, plan on upgrading or replacing the equipment purchased or developed during the second year, etc.

The technology cannot be controlled unless you move with it. Planned maintenance, phasing out and replacement will enable managers of instructional technology to control the implementation and growth. Such control will permit the U.S. to adapt to the structural changes that technology has created within its economy.

Preparing for a Change

The National Commission for Employment Policy, in its report on 'Computers in the Workplace,' stated that computers and computer based equipment are new technologies that are changing what we do and how we do it. Whether the process is termed 'revolutionary' or 'evolutionary' is largely a matter of personal and historical perspective rather than fact. If your job is abolished or dramatically altered when new equipment is introduced into the workplace, it is a personal revolution, even if you are retrained to work with the new equipment or can move to another occupation. From a broader perspective, however, technological change is a continuous process, and the diffusion of computer based education appears more evolutionary than has been the case with other innovations.

In contrast to cyclical changes, which tend to be reversed as economic conditions vary, worldwide computerization is one of the forces causing permanent structural changes in the U.S. economy. A useful analogy is the distinction between changes in the weather (cyclical) and changes in the climate (structural).

There exists strong evidence indicating that the economic climate of the U.S. is changing because of its inability (or unwillingness) to advance new technologies, such as CBT and cognitive systems, to a state of unchallenged excellence. One way in which nations develop a competitive edge is through the intellectual development of their people. An important part of developing people lies in sharpening basic skills and helping individuals to prepare for changes that are due to advances in technology. One way of achieving both objectives is to use advances in technology as part of the individual's training in both schools and industry.

Most nations realize that the rapid change in technology needs to be managed not reversed. Unfortunately, some powerful individuals in some nations are resisting the changes instead of managing them. If a nation expects to remain competitive in the world market, it must provide aid to industries and individuals that are trying to make the transition to the new technologies. Those aided, however, have a responsibility to make conscientious use of public and private resources to prepare themselves for the new environment into which they are moving. One of the best means of successfully making that change is by using technologies such as CBT to maximize both education and training effects. In the process, they will be developing a new industry for the information age.

John J. Hirschbuhl
The University of Akron
Computer Based Education
Akron, OH 44325
(216) 375-6507

Bibliography

Austin, S. (1985, June). Mixed grades for for training software. Computing Systems

Barr, A., & Geigenbaum, E. (1982). The handbook of artificial intelligence (Vol. 2). Los Altos, CA: Kaufman.

Clark, S. (1985, October 10). Electronic publishing has become the jewel of the crown. Computer Weekly.

Feigenbaum, E., & McCordack, P. (1983). The fifth generation. Menlo Park, CA: Addison Wesley.

Feigenbaum, E., & McCordack, P. (1983). The fifth generation: Artificial intelligence and Japan's computer challenge to the world. Reading, MA: Addison Wesley.

Hall, K., Comer, R., & Merrill, J. (1982). Taxonomy of instructional strategies for computer based education. Proceedings: Association for Educational Data Systems.

Hunt, H. A., & Hunt, T. L. (1985). An assessment of data sources to study the employment effects of technological change. In Technology and employment effects--Interim Report. Washington, DC: National Academy Press.

Kearsley, G., Hunter, B., & Seidel, R. (1983, February). Two decades of computer based instruction projects: What have we learned? Technological Horizons in Education.

Nilsson, N. (1980). Principles of artificial intelligence. Palo Alto, CA: Tioga.

O'Neal, F. (1984, May). An alternative model of computer assisted instruction in an educational environment. T.H.E. Journal, 11(8), 113.

Reiser, R., & Gagne, R. (1983). Selecting media for instruction. Englewood Cliffs, NJ: Education Technology Press.

Rich, E. (1983). Artificial intelligence. New York: McGraw-Hill.

Rogers, J. (1984, March). Something has to give; A formula approach to CBT constraints. Data Training.

Zientara, M. (1982, October 11). Control seen as key element in user training. Computer World.

HIDDEN DECISIONS IN THE CBT DESIGN PROCESS

Adele Suchinsky & David Losleben

Department of Defense

ABSTRACT

This paper provides an insight into the need for decision making that arises during course design of a computer-based instructional package. The initial difficulties that a design team will have are reduced with a well thought out planning stage. Even with the best planning, however, the variables of semantics, artistic ability, creativity, and the inconsistencies between hardware and software will continue to present problems. As those problems are resolved, package standards evolve. The factors which make a CBT package dynamic are directly related to the decision points that raise design questions.

INTRODUCTION

Many of the decisions of a Computer Based Training project are obvious and are generally made during the initial planning stage. Once development begins, however, it quickly becomes apparent that more decisions are necessary. For the new CBT designer, decision points can seem to appear at every turn of the page. Often a question is resolved without the realization that a decision has even been made. These decisions, no matter how subtle, will establish the standards that all succeeding lessons in the training package will follow. Frequently, these decisions are not thought about until you get into the actual design work for the lesson. Although up-front design planning is complete, decision points will surface which determine how the training package will appear. It becomes necessary, therefore, to be aware of decisions being made, and the impact these decisions will have on the development of the total training package.

SITUATION ANALYSIS

Picture this scenario: The design team is established and the task analysis has been completed. Behavioral objectives are written, and a course map created. A design model for the development of the lessons is complete. Function keys have been assigned and their operation determined. An authoring package has been selected and team members have familiarized themselves with its capabilities. Even the presentation colors have been selected. As a team member, you feel well organized and prepared to begin designing your first lesson. Which lesson will be designed first? Easy choice, but is it? Should the easiest lesson be designed first or the most complex?

There will be places during the design process where you will have to determine your direction. Few first time CBT designers, however, are aware of the vast number of decision points that await them throughout the design process of that first lesson.

The selection of color, for example, seems at first so simple. "Let's have white letters on a blue background." You try it. It looks good. Decision made...you think! Will you use white letters on a blue background throughout the entire lesson, or will the testing section be different? Should different colors be used for the practice activity or help section? What about color selection for the option keys? Will there be users on a monochrome monitor? Which monochrome monitor will you need to plan for: green? amber? both? How do your color selections appear on these monitors? Are there hardware considerations that need to be addressed now? Have we returned to the pre-design drawing board?

Another seemingly simple topic for decision making is the testing section. The test may be obvious. But where will the test be placed? Should it be at the end of each lesson, or at the end of each unit? How many questions make an appropriate test? What types of test questions should be asked? Should feedback be provided for each question? Can students skip a question and return later? Will there be an option for students to change or verify their answers? What percentage of correct answers will constitute mastery? Where does the student go if s/he fails the test? Should directions for the tests be consistent from one lesson to another? Does the authoring system limit the types of questions possible? Is there a choice to be made between performance tracking and an option to branch based on student performance? Which one should be selected? Did the pre-design drawing board cover these possibilities?

CREATING THE STANDARDS

Many CBT design decisions are made without realizing that a decision has been made. Others we agonize over for a long time before reaching a decision. Most decisions will be made during the design phase of the first lesson selected. These decisions will influence the design, format, and flow of the lessons to follow.

There is no particular place where it can be said that a decision point stops development. The points appear everywhere during the design of the project. There are, however, a few broad categories where problems of a similar nature can be found. These are the areas where course design standards can be determined: Structure of the Training Package, Screen Design, and Graphics.

The TRAINING PACKAGE structure has perhaps the largest impact on the student and learning. Issues currently being researched regarding student learning with a computer can be found here. You have the material you want to teach, but what you do to enhance the information being presented affects the package structure. The following determinations set the guidelines for development of additional lessons, and assure consistency of the training:

*placement of tests, frequency of testing, number and type of items per test

*modifications to directions to avoid monotony yet assure consistency

*availability of on-line help such as glossaries, examples, course map, and option keys

*control over the direction of learning within the lesson and the module, and the frequency and location of student quit options

SCREEN DESIGN encompasses decisions made concerning the visual that the student sees:

*the amount of screen space used for text

*chunking of information on the screen
    blocking of information
    use of open space
    placement of graphics
    use of windows

*utilization of color
    text
    background
    highlighting
    borders

*cueing strategies
    placement
    style
    frequency

When dealing with GRAPHIC ISSUES, there needs to be a balance between system capabilities and artistic execution. Flexibility with graphics is a necessity, as is the communication between the graphic artist and the instructional designers. What the designer envisions may not be what the user wants or the authoring system is capable of doing. Issues concerning graphics include:

*utilization of screen space, especially when text must be included

*compromises in design that need to be made to be compatible with the capabilities of the authoring system or graphics packages used

*selection of color depending on the type of monitor(s) to be used and changes which take place if graphics are imported

*level of acceptability if animation is desired

SAMPLE DECISIONS

Let's look at some typical decision points for the first lesson of a tutorial. Some may be decided for you by the capabilities or limitations of your authoring system. Others may be determined by the wishes of the user. The list is presented in no particular order of priority, nor are they necessarily the ones you need to resolve for your training package.

If you are going to include help sections such as a glossary or help, where will the student be given the options to access these areas?

How often do you test with items tracked by item analysis?

What number of test questions should be presented per test, and should this number be consistent throughout the training package?

Where will you use graphic title pages or decide to use a text cueing strategy instead?

Should you number the pages of the lesson? If so, where?

How many sub-tasks can be effectively placed within one objective?

Should the lesson objective be verified as to the student's understanding of what he is about to learn?

Should each multiple choice answer receive its own feedback?

What will the extent of the branching be for each incorrect response?

To what degree can test directions be reworded from one lesson to another to avoid monotony, yet still be consistent for the learner?

If you are offering the student a second chance to take the lesson test, will he be given the same test or a different one?

When the student has failed the test for the second time, where will you send him?

Is "OJT" an option if the student does not pass the lesson test? module test? training package test? At what point do you offer the student another training option?

How many times will you let the student take a test?

What kind of test feedback will you give the student...final score? individual question feedback? correct answers or merely "right" or "wrong"?

What percentage of correct answers on a test will determine student mastery? Does this have to be consistent throughout the entire training package?

Should the student have the option of changing his answers on a test before they are logged? Is this option worth the time it will take to code?

Where will the students have the option of returning to the main menu and making another lesson selection? Does s/he need to complete one lesson before selecting another? Can he move around freely? Are all lessons dependent upon one another in a hierarchy or only some?

If you are down loading information from a mainframe, what amount will be down loaded at a given time... lesson or module?

Where will student "quit" options be placed? At the end of an activity? At the end of the lesson? Both?

When simulating an actual computer screen, how do you differentiate the simulated screen from the tutorial explanation which accompanies it?

What learner control options will be available and when?

Should title pages be templated so they all look alike in each lesson?

How much highlighting of terms is reasonable without being distracting?

Will lessons or modules completed by the student be indicated?

If you had to choose between "branch on score" or "item analysis", which one would you choose?

How much interactivity should an "Introduction" have?

When the student signs on after the initial time, where will he go? How much will he bypass?

What types of "cueing strategies" will be used through out the training to let the student know where he is? How many different types of cueing strategies will you use? Where will they be placed? How often will they be provided?

Will question feedback be general or item specific? Will you offer help or hints? Will feedback be simple or complex? Will student responses be timed?

SUMMARY

No matter how many decisions you make in advance, you can't make them all. Most of the "nuts and bolts" decisions are made during the design phase of the first lesson designed. The suggested decision points you have just seen may or may not be the same for your instructional package. The authoring system, subject matter, design model, and expertise of your design team are all variables at work during the development of your training package. When you plan a program, you can't think ahead to all the details such as where question responses will be placed. And yet, this is a decision that will become a standard for your training package to insure consistency throughout the presentation.

There will always be some questions that will not arise until you are considering the details in depth during the initial design process. Strive for consistency without monotony, and a high level of interest without compromising the instructional effectiveness.

Hidden decisions are going to have an effect on scheduling and projected completion dates for your package. Some decisions require time and experimentation, while others may necessitate an alternative direction. No matter how much time you put into pre-planning, more decisions will be made which are precedent setting and lead to other decisions which are precedent setting, and so on. The first lesson could take up to 1000 hours for a novice CBT team. The time and effort expended will reap rewards in succeeding lessons where these "hidden" decisions have become standards.

DESIGNING SOFTWARE FOR EASY TRANSLATION INTO OTHER LANGUAGES

Stanley R. Trollip - S.R.Trollip & Associates

Gary Brown - Media Computer Enterprises

Abstract

Software applications are being exported more and more to countries whose language is not English. Often successful penetration of these markets depends on translating the application into the language of the country. This paper discusses the need for incorporating into the initial design of the software the means to make such a translation easy. A software package developed by the authors and later translated into Dutch is used as an example.

Introduction

If one travels to other countries in which English is not the native tongue, one is struck by the fact that most computing is still English based. Most programming languages have an English-language basis and most people working in the computer industry speak English to some degree.

Although this situation is often not productive from a production perspective, it has not historically particularly affected the users of computer programs, because what they see is usually written in their native tongue. That is, the inconvenience rests mainly with the implementor of the program, not the user.

Some countries have circumvented this implementation problem by translating the commands of the programming language from being English-based into the local language. Because of the limited number of commands in a programming language and the fact that they are strictly defined, this is a relatively easy task. As micro-computers have spread, however, so has the volume of packages that have a broad appeal, such as word-processors and spreadsheets.These are consumer packages, and hence the foreign consumer is being exposed more and more to the situation of having to be functional in English in order to use the software effectively and efficiently. For a non-English speaker, this can be extremely frustrating, particularly when the manuals are also not translated.

From a marketing perspective if a software application has international sales' possibilities, it maximizes the chances of success if the consumer can use the software in the local language. This paper discusses ways to design software to minimize the problems associated with translating what the users see when using a software package from English into a foreign language.

We use as a vehicle for our discussion a package we designed and implemented that enables the user to create and administer examinations. The package is called The Examiner (Trollip & Brown, 1986). It is currently available in English and Dutch, and is process of being translated into French and Spanish.

Overview of The Examiner

The Examiner Testing System is a sophisticated exam creation and administration system for use on IBM compatible PC computers. Its purpose is to provide organizations with the capability of implementing exams to be administered on paper or via the computer without having to make use of programming expertise. Particular care has been taken in the design to ensure that the user interface is clear and unambiguous to the novice, while not hindering the experienced user. For the most part the design criteria discussed in Alessi and Trollip (1985) were followed.

The Examiner comprises three parts or subsystems: the Item Editor; the Examination Editor; and the Test Administration subsystem. All subsystems are completely integrated. The Item Editor allows the user to define implement, amend, or view the item database. The Examination Editor permits the user to define the nature of examinations to be administered. The Examination Administration subsystem is the software that administers the examination to the examinee via the computer.

There are a couple of features worth mentioning. Throughout the software, context-specific online help is available. That is, at any time the user can request more information about the currently available options merely by pressing a key. And secondly, the user is shielded from the operating system at all times. Such things as checking whether disks have been formatted and changing directories and subdirectories are accomplished without leaving the system.

There are two distinctly different users of the system - the instructor who interacts with the Item Editor and Examination Editor to create exams, and the student who takes the exam. Even if the instructor is fluent in English, it is unlikely that the examinees will be.

As we planned the system, we decided that we wanted to access an international, multilingual market. From the outset, therefore, we designed the system to be easily translatable. Furthermore, we speculated that the translation was likely to be accomplished by native speakers in their own country. So, we also had to design methods by which minimum ongoing support was needed by the translator. Finally, in order to avoid duplication of effort, we wanted the translator to do most of the implementation of the new version, with us doing only necessary fine tuning.

Anticipated Problems

As the planning process proceeded, a number of problems were anticipated that would occur when the system was translated into other languages. These problems would occur both during implementation and conversion.

1. We wanted one set of computer programs to handle all languages. A separate program for each language would have led to an unwieldy and error-prone support problem. Therefore, if one set of programs was to serve all languages, the program itself would have to know which language was being used.

2. Any method used for implementing the multilingual system could not increase the binary size of the program too much.

3. To speed up the translation process, particularly since translation of the system would most likely take place outside of the United States, it was important to construct the system so that the person doing the translation could implement most of the changes without having to know programming.

4. Displays comprise static information that never changes, and variable or transient information, whose contents depend on the particular implementation. As an example of the latter, different users will have different descriptors for parts of the database, different subdirectory names, and different numbers of items in the database.

   The problem here is that static information usually varies in length from one language to another, making it impossible to place transient information in a fixed location on the screen.

5. There are cultural differences between languages that need to be taken into account, such as how dates and decimal numbers are written.

Solutions to Anticipated Problems

The first three problems above were partially solved by a single design decision, namely to separate the body of the program from the text that appears on the screen or that is sent to the printer. In essence, when in operation the main Examiner program constantly accesses a database of text messages that is separate from the program itself.

1. By placing the text in a separate database, different language versions of the program can be presented using the same program and different text databases.

2. By implementing this scheme, the size of the binary was virtually the same for the multilingual version as for a single-language version.

3. By having the text in a database, it was possible to construct a simple editing system for the translator to use. The editor displayed the English text and the translator was told to make the translation directly on the screen, preserving the original screen layout as closely as possible. The translator was also instructed to note whenever the translation was unable to preserve the original format. This was particularly important when transient information was associated with the text.

   Once the translation was completed, the program developer went through all the displays noted as exceptions by the translator and checked whether they were displayed properly during program execution. When problems were found, a further attempt was made by the translator to produce a translation that was closer in size to the original text, or an exception was coded into the main program to adjust the position of transient information on the screen to match the underlying static display.

   This exception was activated within the program by referencing what language was being used. What this meant was that the main program grew in size only to cater to exceptions.

4. In order to ensure that transient information appeared in the same position on the screen irrespective of the language used, it was never embedded in the middle of a sentence if at all possible. So instead of displaying There are 345 items in the physics database.

   The following convention was used:
   Database: physics
   Number of items: 345

   In this way, it did not matter how long the foreign language translations were for Database and Number of items. The position of the database name and the number of items remained fixed.

   In choosing this approach, there was some sacrificing of fluency of language for uniformity of presentation. On some displays, too, there is a lot of blank space between static and transient information to allow for 'worst-case' translations. However, considering the binary size savings that can be realized, this trade-off is reasonable.

5. In the United States, dates are written in numeric form as follows:
   3/9/1947 is equivalent to 9th March 1947.

   In most other countries this date would be interpreted as 3rd September 1947.

   The obvious solution to this cultural difference is to use words for months and not numerals. So throughout The Examiner the convention used is 9 March 1947.

In similar fashion, most countries write the number "three point one five", for example, as 3,15 in contrast to 3.15 in the United States.

The solution to this is to allow commas or periods to be used interchangeably in instances such as answers to open-ended numeric items.

Implementation

The first test of the design came about at the end of 1986, when the Dutch software publisher H.E. Stenfert Kroese bought the publication rights to a Dutch-language version of The Examiner. A printout of all the English displays was sent to the translator, together with the display database editor and the English text database. The translator was given the instructions mentioned above to try and fit all translations within the length of the English text and to note all exceptions. The complete translation was finished within 6 calendar weeks.

In January 1987, the second author, who had programmed The Examiner, spent approximately one week installing the translated database to work with the main program. In a few places this necessitated modifying the main program to cater to the exceptions that could not be translated within the desired constraints or working with the translator to develop a more suitably formatted translation.

Most of this time was actually spent making modifications to the main program where it did not meet the design guidelines. These violations for the most part consisted of a few hard-coded messages that responded to unusual program conditions, or sections of code that had been produced very early in the development process. With these further modifications in place, there are now no hard-coded text messages at all within the main program. It is anticipated that the implementation of the French translation will take even less time.

Conclusion

The original design of The Examiner to cater to translation into different languages has been very successful. Very little effort was necessary in modifying the main program to make The Examiner work in Dutch.A major benefit of separating the text from the program and allowing a translator work directly with the text database is that each language version acquires much of the cultural flavor of the language. Thus the Dutch version does not look like just a translated version of the original American- English system, but is really a Dutch product - that feels as if it were produced entirely in the Netherlands.

Our experience has been that if proper consideration is given during the design process to the possibility of foreign language versions of software, it is possible to implement them relatively easily.

References

Alessi, S.M. & Trollip, S.R. Computer-based Instruction: Methods and Development. Englewood Cliffs, NJ: Prentice Hall, 1985.

Brown, G.C. & Trollip, S.R. (1987) The Examiner (Dutch language version). Leiden, The Netherlands: H.E. Stenfert Kroese.

Trollip, S.R. & Brown, G.C. (1986) The Examiner. Mendota Heights, MN: Media Computer Enterprises.

Stanley Trollip
159 Pillsbury Drive SE
University of Minnesota
Minneapolis, MN 55455
(612) 625-1323

DEVELOPMENT OF A MICRO-COMPUTER BASED KNOWLEDGE SYSTEM: AN OVERVIEW

Alan D. White

Integrated Communications Technology, Inc.

ABSTRACT

This study addresses the creation of a micro-computer based knowledge system software package to assist trainers and educators in the preparation of behavioral learning objectives. Most trainers and educators are unfamiliar with knowledge system concepts, capabilities and constraints. The product outcome from this study should provide trainers and educators with a usable, hands-on knowledge system tool.

INTRODUCTION AND STATEMENT OF THE PROBLEM

Using a micro-computer small knowledge system building tool, this study effort addressed the creation of a stand-alone knowledge system designed to assist instructional developers in the preparation of behavioral learning objectives. A small knowledge system software shell was selected based on having an inductive front end with a powerful rule-based system, an ability to connect to other micro-computer software, being easy to use and having reasonable cost.

Knowledge system software is programmed to solve small, difficult problems. Applications most suited to this technology are consultation in a narrow but knowledge-rich area of expertise. Knowledge system programs have considerable potential for training and education and accordingly should become increasingly familiar to instructional specialists (Harmon, 1986; Zemke, 1983).

One knowledge domain in the area of instructional systems where expert opinion is reasonably well-defined is the preparation of behavioral learning objectives (Bloom, 1971; Bloom, 1977; Gagne, 1970; Gagne, 1974; Gronlund, 1970; Kibler, 1981; Mager, 1968; Mager, 1972; Pipe, 1974). Since research has shown that the more successful knowledge system software projects are classification models (Berry, 1986; Hayes-Roth, F., 1983) in which a body of data is analyzed for prescribed solutions, the preparation of behavioral learning objectives appeared to be a logical candidate for the application of this technology.

The critical first step in instructional design, assuming instruction has been selected as the appropriate method to bring about changes in performance, is the preparation of clear, concise, measurable instructional objectives. Well written behavioral objectives are the foundation for the preparation of instruction, the selection of methods of instruction, the determination of subject matter content, and the assessment of the outcomes of instruction. The objective clearly defines where one wants to go. Once the objectives are written, one can create and administer the means of getting there and finding out whether one has arrived.

A well written instructional objective describes (Mager, 1975):

o Behavior that is clearly measurable or observable.

o The learner performance that is to be evaluated at the end of instruction.

o The conditions under which performance following instruction will be assessed.

o The performance level that is considered a reasonable indicator of mastery.

The focus of this study effort was selected because of the limited research that has been conducted addressing the application of knowledge systems to the fields of training and education. Although there have been extensive developments in industry, medicine, and the Department of Defense, there appears to be a comparative dearth of research efforts applying knowledge systems to the problems in public education (Sleeman, 1982). One report estimated that fewer than a half dozen doctoral theses have appeared each year in some aspect of building knowledge systems (Stefik, 1982). Since small knowledge systems will prove to be an appropriate solution for a large number of problems faced in training and education, ideally instructional technologists will be among the leaders in applying this new technology (Hofmeister, 1986).

Correspondingly, one of the byproducts of building a knowledge system is the refinement of one's personal understanding of the techniques of developing behavioral objectives. The knowledge system development process forces one to analyze systematically the behavioral learning objective writing process. The analytical and mental processes involved tend to sharpen thinking and are frequently found to be professionally enlightening (Fersko-Weiss, 1985).

IMPLICATIONS FOR THE IMPROVEMENT OF EDUCATIONAL PRACTICE

Heretofore two factors have contributed to the delay in use of knowledge systems in training and education. First, there has been a lack of suitable computer equipment for knowledge system development (Brady, 1986; Ferrara, 1985). Knowledge systems have traditionally been developed on mainframe computers, usually not within easy access to the trainer and educator. Second, traditional development has required the use of expert programming languages such as LISP and PROLOG. The potential use of knowledge systems in training and education has now changed, however (Hofmeister, 1986; Kearsley, Greg, 1987; Lawlor, 1984; Tennyson, 1987; Tilmann, 1984). Improvements in microcomputer hardware coupled with the development of micro-computer based knowledge system authoring tools have made the use of these programs appear much more feasible today (Hofmeister, 1986).

In each occupational field, in the early stages of knowledge system developmeant and acceptance, it is important to convey knowledge system concepts, capabilities and constraints. Most trainers and educators are unfamiliar with these basic concepts. They do not have significant appreciation as to how knowledge systems can be used in practice, what benefits can be achieved, and what limitations are present in specific applications. Hands-on experience is important in gaining this appreciation (Coombs, 1984)

METHODOLOGY

The steps required in the development of a small knowledge system are described in detail by Harmon (Harmon, 1985). The six steps that one must go through are summarized as follows:

1. Select a tool and implicitly commit to a particular consultation paradigm.

2. Identify a problem and analyze the knowledge to be included in the system.

3. Design the system.

4. Develop a prototype of the system using the tool. This includes actually creating the knowledge base and testing it by running enough consultations.

5. Expand, test and revise the system until it does what it is supposed to do.

6. Maintain and update the system as needed.

VALIDATION

To provide an objective validation of this study effort, a validation instrumeant was prepared based on the knowledge system validation work of Sells (Sells, 1985). Criteria used in the instrument were knowledge system consistency, completeness, soundness, precision, and usability. The knowledge system software product along with the questionnaire was provided to two professional colleagues from the American Society for Training and Development and one professional colleague from the National Society for Performance and Instruction; three Doctor of Arts in Training and Learning Technology colleagues enrolled at Nova University; and two faculty members from Nova University. Results from this validation effort have been incorporated into the final version of the knowledge system product.

RESULTS

The knowledge system package, WRITING BEHAVIORAL OBJECTIVES was designed to guide users in understanding the components of behavioral objectives---a performance, a condition, and a standard---as well as assist users in writing knowledge, comprehension, application, analysis, synthesis and evaluation objectives. Also, users are assisted in classifying existing objectives obtained from other sources according to the action verb used in the wording of the objective.

The WRITING BEHAVIORAL OBJECTIVES knowledge system consists of four separate knowledge bases named INTRO, VERBS, DEPTH, and BLOOM. Each one is individually accessed and consulted using the copyrighted Paperback Software package, VP-EXPERT.

INTRO is an introduction and review of behavioral objectives. Users can review the purposes served by behavioral objectives; behavioral objective components---a performance, a condition and a criterion; and brief definitions of each of the six levels of behavioral objectives---knowledge, comprehension, application, analysis, synthesis, and evaluation.

VERBS prompts the user in selecting behavioral objective components---a condition, a criterion and a performance. The user is advised as to what level of learning the objective is addressing based on the user's choice of an action verb. This knowledge base also can be used to help classify the learning level addressed by an existing behavioral objective.

DEPTH provides the user with an opportunity to prepare behavioral objectives to the specific learning level desired. The user starts by selecting a condition and a criterion. Next the user is asked to identify the category of behavioral objective to which he/she wishes to write. Finally, the user selects an action verb appropriate to the learning level of choice.

BLOOM material was adapted from TAXONOMY OF EDUCATIONAL OBJECTIVES: The Classification of Educational Goals: HANDBOOK I: COGNITIVE DOMAIN by Benjamin S. Bloom et al. New York: Longman Inc., 1956. Examples of objectives written to various learning levels can be accessed.

REFERENCES

Berry, Dianne C., et al. (1986). "Expert Systems and the Man-Machine Interface," EXPERT SYSTEMS. 228-231; October.

Bloom, B.S. (1971). HANDBOOK ON FORMATIVE AND SUMMATIVE EVALUATION OF STUDENT LEARNING. New York, NY: McGraw-Hill.

Bloom, B.S., et al. (1956). A TAXONOMY OF EDUCATIONAL OBJECTIVES. HANDBOOK I: THE COGNITIVE DOMAIN. New York, NY: Longman.

Brady, Holly (1986). "Artificial Intelligence," CLASSROOM COMPUTER LEARNING. 16-29; January.

Coombs, M. J. (Ed.)(1984). DEVELOPMENTS IN EXPERT SYSTEMS. New York: Academic Press.

Ferrara, Joseph M., et al. (1985). "Expert Systems Authoring Tools for the Microcomputer: Two Examples," EDUCATIONAL TECHNOLOGY. 39-41, April.

Fersko-Weiss, Henry (1985) "Expert Systems: Decision-Making Power," PERSONAL COMPUTING. 97-105; November.

Gagne, R.M. (1970). THE CONDITIONS OF LEARNING, 2nd Edition. New York, NY: Holt, Rinehart & Winston.

Gagne, R.M., & Briggs, L.J. (1974). PRINCIPLES OF INSTRUCTIONAL DESIGN. New York: Holt, Rinehart & Winston.

Gold, Jordan (1986). "Do-It-Yourself Expert Systems," COMPUTER DECISIONS. 18 (2) 76-81, 14 July.

Gronlund, N.E. (1970). STATING BEHAVIORAL OBJECTIVES FOR CLASSROOM INSTRUCTION. New York, NY: Macmillan.

Harmon, Paul & King, David (1985). EXPERT SYSTEMS: ARTIFICIAL INTELLIGENCE IN BUSINESS. New York: John Wiley & Sons, Inc.

Harmon, Paul (1986). "Expert Systems, Job Aids, and the Future of Instructional Technology," PERFORMANCE AND INSTRUCTION. 25(2), 26-28, March.

Harmon, Paul (1986). EXPERT SYSTEMS STRATEGIES: The Monthly Newsletter for Managers and Developers of Expert Systems. 2 (9), September.

Hawkins, William J. (1986). "Expert Systems Promise Supersmart PCs," POPULAR SCIENCE. 83-86: March.

Hayes-Roth, Barbara, et al. (1985). "Paradigms for Intelligent Systems," EDUCATIONAL PSYCHOLOGIST. 20(4):231-241.

Hayes-Roth, Frederick, Waterman, Donald A. and Lenat, Douglas B. (Eds)(1983). BUILDING EXPERT SYSTEMS. Reading, MA: Addison-Wesley Publishing Company.

Hofmeister, Alan M. and Ferrara, Joseph M. (1986). "Expert Systems and Special Education," EXCEPTIONAL CHILDREN. 235-239: November.

Kearsley, Greg P. (Ed)(1987). ARTIFICIAL INTELLIGENCE & INSTRUCTION. Reading, MA: Addison-Wesley Publishing Company.

Kibler, R.J. (1981). OBJECTIVES FOR INSTRUCTION AND EVALUATION. Boston: Allyn and Bacon.

Lawlor, Joseph (1984). ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS. Los Alamitos, CA: Southwest Regional Laboratory for Educational Research and Development, 15 January.

Mager, R.F., & Beach, K.M. Jr. (1968). DEVELOPING VOCATIONAL INSTRUCTION. Belmont, CA: Fearon Publishers.

Mager, R.F. (1972). GOAL ANALYSIS. Belmont, California: Pitman.

Mager, R.F. (1975). PREPARING INSTRUCTIONAL OBJECTIVES. (Second Edition). Belmont, CA: Pitman.

Pipe, P. (1974). OBJECTIVES FOR CHANGE. Belmont, CA: Pitman.

Sells, Peter S. (1985). EXPERT SYSTEMS: A PRACTICAL INTRODUCTION. New York, NY: John Wiley & Sons.

Steelman, D. and Brown, J. S. (1982). INTELLIGENT TUTORING SYSTEMS. London: Academic Press.

Stefik, M., Aikins, J., Balzer, R., Benoit, J., Birnbaum, L., Hayes-Roth, F., & Sacerdoti, E. (1982). "The Organisation of Expert Systems: A Tutorial." ARTIFICIAL INTELLIGENCE. 18 (2), 135-174, March.

Tennyson, Robert D. and Ferrara, J. (Eds)(1987), "Special Issue: Artificial Intelligence in Education." EDUCATIONAL TECHNOLOGY. 27 (5), May.

Tilmann, Martha J. (1984). ARTIFICIAL INTELLIGENCE AND ITS IMPORTANCE IN EDUCATION. Michigan State University.

Williamson, Michey (1986). ARTIFICIAL INTELLIGENCE FOR MICRO-COMPUTERS: THE GUIDE FOR BUSINESS DECISIONMAKERS. New York, NY: Brady Communications Company.

Zemke, Ron (1983). "Artificial Intelligence: What Trainers Can Learn From the Silicon Frontier," TRAINING. 65-91: December.

APPENDIX A
SMALL KNOWLEDGE SYSTEM VENDORS

Airus, Inc. (Airus-A)
10200 S. W. Nimbus Ave., Suite G-5
Portland, OR 97223
(503) 620-7000

California Intelligence (Xsys)
912 Powell Street
San Francisco, CA 94108
(415) 391-4846

Exsys, Inc.(Exsys)
Box 75158, Contr. Sta. 14
Albuquerque, New Mexico 87194
(505) 836-6676

Expert Systems International
(ES/P Advisor)
1150 First Avenue
King of Prussia, PA 19406
(215) 337-2300

Human Edge Software Corp.
(Expert Ease, Expert Edge)
1875 S. Grant Street
San Mateo, CA 94402-2669
(415) 573-1593
(800) 624-5227

KDS Corp. (KDS System)
934 Hunter Road
Wilmette, IL 60091
(312) 251-2621

Level Five Research, Inc.
(Insight 1+ and Insight 2+)
503 Fifth Avenue
Indialantic, FL 32903
(305) 729-9046

Miller Microcomputer Services (EX-PERT-2)
61 Lake Shore Road
Natick, MA 01760
(617) 653-6136

Neuron Data (Object Nexpert)
444 High Street
Palo Alto, CA 94301
Attn: Mr. Patrick Perez
(415) 321-4488

Paperback Software (VP Expert)
2612-8th Street
Berkeley, CA 94710
(415) 644-2116

Jeffrey Perrone and Assoc.,Inc.
(Expert-Ease)
3685 17th Street
San Francisco, CA 94114
(415) 431-9562

Programming Logic Systems
(Micro-Prolog Professional)
31 Crescent Drive
Milford, CT 06460
(203) 877-7988

Radian (Rulemaster)
8501 Mo-Pac Blvd.
P.O. Box 9948
Austin, TX 78766
(512) 454-4797

Systems Designers Software
(VP Expert and Softsync)
444 Washington Street, Suite 407
Woburn, MA 01801
Attn: Ms. Jacquie Dowsette
(617) 935-8009

Teknowledge Inc. (M.1)
525 University Avenue
Palo Alto, CA 94301-1982
(415) 327-6600

Texas Instruments Incorporated
(Personal Consultant)
P.O. Box 1444, M/S 7722
Houston, TX 77251
(512) 250-7047
(800) 527-3500

Thoughtware Inc. (Management Advantage)
2699 South Bayshore Drive
Suite 1000A
Coconut Grove, FL 33133
(305) 854-2318
(800) 848-9273

Contact: Alan D. White
Integrated Communications
Technology, Inc.
2807 Fort Scott Drive
Arlington, VA 22202-2306

703-549-2807

On-Line Tools for 3-D Graphics Programming

Mason Woo

Silicon Graphics Inc.

ABSTRACT

Teaching 3-D, interactive, color graphics defies traditional methods. The two-dimensional printed page or blackboard doesn't provide depth; its content isn't rich enough. A series of on-line instructional tools (IRIS Graphics Labs) link the visualization of 3-D Graphics and the mathematical values of a graphics programming routine. These tools are a proven part of the IRIS graphics training curriculum.

CR Categories and Subject Descriptors: D.3.3 [Programming Languages]: Language Constructs — *Procedures, functions and subroutines*; I.3.4 [Computer Graphics]: Graphics Utilities — *Graphics packages*; K.3.1 [Computer Milieux]: Computers and Education — *Computer-assisted instruction (CAI)*

Key Words and Phrases: three-dimensional viewing, interaction, animation

1. Introduction--Animating Graphics Routines

Applications using computer graphics will become more sophisticated, and will incorporate three-dimensional geometry, color, and real-time interaction. Instructors will discover two-dimensional media does not explain three-dimensional concepts. 3-D can best be understood through 3-D media.

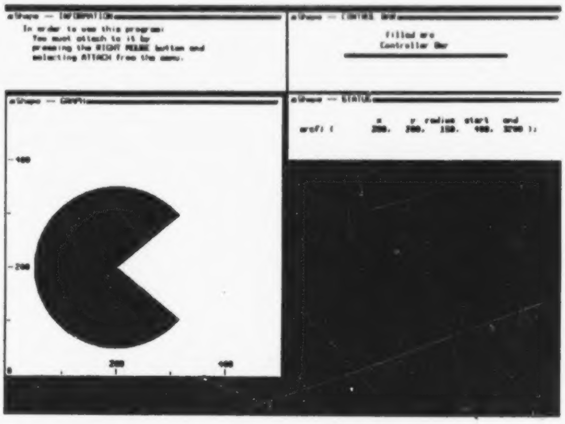
To create applications for the IRIS graphics superworkstation, a programmer uses the IRIS Graphics Library. A series of computer-based, educational tools, called *Graphics Labs*, allows a student of graphics programming to interactively animate a graphics library routine. The Graphics Lab connects the graphics routine syntax to the visible object that it produces on the screen. The student can directly experiment with the mathematical values of the parameters of a graphics routine.

There are 16 Graphics Labs that illustrate a wide range of graphics concepts, including two-dimensional and three-dimensional geometry, color and shading. Where possible, the layout and interaction among different Graphics Labs are similar. User interface operations with one Graphics Lab are generally transferable to other Labs.

The following sections discuss the implementation of a Graphics Lab and examine the specific problem of visualizing three-dimensional geometry.

2. Graphics Lab--Display, Use and Features

A Graphics Lab opens several windows on the high-resolution screen of the IRIS superworkstation. The student uses a three button mouse for interaction with different types of windows: *help, status, controller*, and *view*. The Graphics Lab for experimenting with two-dimensional shapes is shown below.



In order to use this program:
You must attach to it by pressing the RIGHT MOUSE button and selecting ATTACH from the menu.

The help window contains operating instructions and error messages. For all Graphics Labs, the help window is in the upper left corner of the display.

The status window contains the syntax and updated values for an IRIS Graphics Library subroutine.

A slider bar in the controller window is used to modify parameters in the status window. The controller window is just above the status window.

View windows illustrate the execution of the subroutine in the status window. A Graphics Lab may have one or many view windows, each from a different point of view.

The student is encouraged to actively experiment rather than passively observe. The operation of a Graphics Lab elucidates how a student understands a graphics subroutine and its meaning. The student uses the mouse to position the cursor over a parameter in the subroutine and then presses the left mouse button to declare his or her intent to modify that value. (The right mouse button governs a pop-up menu which can choose among several graphics subroutines, reset parameter values, or exit the lab.) After selecting a parameter value, the student uses the controller bar to modify it. The student uses the left mouse button to select and move the tab on the controller bar. If he or she moves the tab off the end of the controller bar, it wraps around; a feature inspired by the number wheel device. [7] In the status window, the changing numerical value in the subroutine is displayed and updated. The status window is akin to a debugger which displays variables and their contents.

Several view windows pictorially animate objects as they change due to the changing parameters given to the graphics subroutines. The view windows illustrate objects from different points of view, some from the eye position of a normal viewer and others from various God's-eye views. To view three-dimensional objects, students essentially need several different views. As the parameter values change, the many views of the scene are continually updated in real-time. Some expert programmers also refer to the Graphics Labs during application development.

The multiple window approach is a variant on Brown University's BALSA [1] instructional environment. For example, the status window is similar to the BALSA *Code View* window. BALSA is directed towards illustrating algorithms and following the the execution of entire programs, whereas Graphics Labs isolate a few lines of a program.

Graphics Labs allow a high degree of direct user control and meaningful reversible actions. The direct subroutine manipulation approach is used for many graphics concepts. The student chooses the value to be animated, stops and reverses the animation, and varies its speed. He or she is constantly reminded of the order and data types of the parameters in the subroutine. The subroutine syntax is depicted in the programming language preferred by the student. C and Fortran subroutines are currently available.

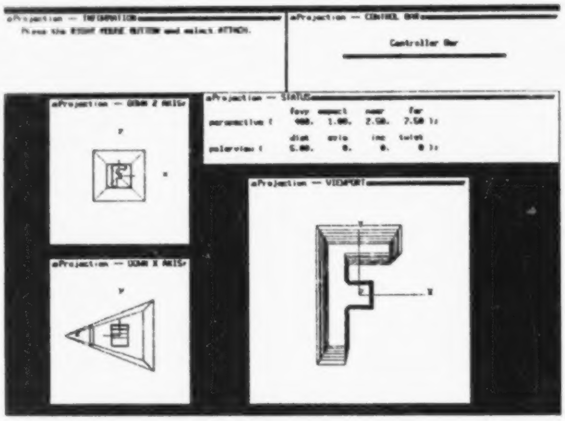
3. Case Study–Viewing Three-Dimensional Geometry

This section examines the effectiveness of Graphics Labs for teaching three-dimensional viewing operations. Three-dimensional graphics requires a different explanation than does two-dimensional. With two dimensions, you envision painting areas on a flat surface. With three dimensions, you must think about an object floating in space and a point of view (position of your eye). First, you set up the boundaries (the viewing volume) around your field of vision; then position the eye and point it in the direction of an object; finally, you project the three-dimensional object onto a two-dimensional monitor. Viewing in three dimensions is analogous to taking pictures with a camera. You select a lens for the camera, set up its tripod, point it at an object, take a picture to create a photographic negative, and enlarge it to get a finished print.

There are two ways to set up a viewing volume. Viewing with perspective projection is the more realistic way. A pyramid-shaped viewing volume outlines the boundaries of a perspective field of vision. The eye is at the center of projection or, more concretely, at the tip of the pyramid (looking towards the rectangular base). With perspective, objects which are further away from the eye occupy less of the viewing volume and appear smaller. Human vision naturally perceives the world with perspective.

The other method is to use a parallel projection, which does not distort sizes. The viewing volume for a parallel projection is a rectangular parallelepiped. Therefore objects of the same size occupy the same amount of the viewing volume. Distance from the eye position has no effect on the perceived dimension of an object. Viewing with a parallel projection appears unnatural but can be useful for drafting, where preservation of size is essential.

A Graphics Lab illustrates the difference between perspective and parallel projection and explains how to write a graphics application which utilizes a projection. In the diagram below, there are three view windows. On the left, two view windows show distinct, God's-eye (once removed) views of the perspective viewing pyramids. Inside each pyramid is an object, an extruded letter F. In the window on the lower right is the view of the letter F from the eye position at the tip of the pyramid



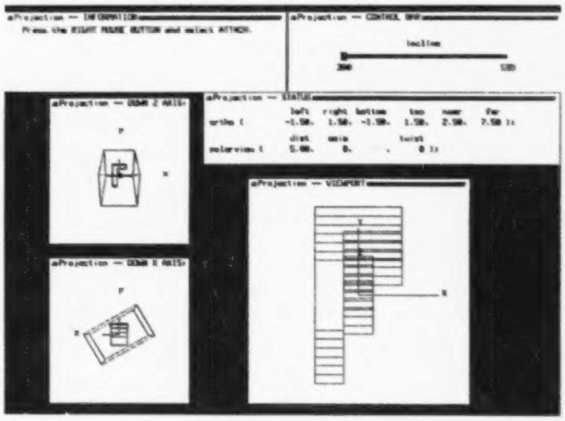
Press the RIGHT MOUSE BUTTON and select ATTACH.

```
                fovy   aspect   near    far
perspective (   400.   1.00.    2.50.   7.50 );
                dist   azim     inc     twist
polarview (     5.00.  0.       0.      0 );
```

In the status window, the IRIS Graphics Library [5] subroutine **perspective** controls the size and shape of the pyramid. The student chooses any of four parameters: *fovy, aspect, near,* and *far*, and change its value with the controller bar. The *fovy* is the width of the field of view in tenths of degrees. For example, the current value of 400 for fovy means the viewer sees 20 degrees to both the left and right of straight ahead, a total of 40 degrees (or 400, when measured in tenths of degrees). The *aspect* is the ratio between height (y-axis) of the pyramid and its width (x-axis). The *near* and *far* values truncate the front and back of the pyramid along the z-axis.

The strength of the Graphics Lab as a learning tool is that the student can control and animate the change in any of these parameters. The parameter can be selected and its value modified with the controller bar. The change can be viewed from the two God's-eye views or from the normal eye position. When the student feels confident with one parameter, he or she may investigate another. The feedback from the experiments is given instantly, and the experiment is entirely controlled by the user. The subject of the experiment is the graphics subroutine, which is exactly what the student is trying to comprehend.

Once the student understands the perspective projection, he or she selects a different projection, the parallel projection. As seen below, the viewing volume for an orthographic parallel projection is shaped as a rectangular parallelepiped, and the object (letter F) is drawn without decreasing size in the distance. (In an orthographic parallel projection, the direction of projection is perpendicular to the projection plane. [3] Think of the projection plane as the film in your camera for your photographic negative.) In the status window, a different graphics subroutine, **ortho** controls the shape of the viewing volume.



Press the RIGHT MOUSE BUTTON and select ATTACH.

```
                left   right  bottom   top    near    far
ortho (        -1.50,  1.50,  -1.50,   1.50,  2.50,   7.50 );
```

During our investigation, we discovered students have very strong preferences about its user interface. Students dislike typing. A graphics subroutine interpreter (for illustrating composited transformations) failed because new students do not yet know what to type, make typographical errors, and need to isolate the actions of one variable at a time. By restricting methods of input, students have a simple set of options and the implementer who creates the learning tool has less to program.

Students preferred to view an asymmetric shape. The extruded letter F was more comprehensible than a cube.

Many of the principles for teaching three-dimensional viewing are also found in BUMPS (Brown University Multiple Projection System).[4]. The key difference between BUMPS and the IRIS Graphics Labs is the manipulation of the single graphics subroutine. Once a student learns this technique, her or she can experiment with other subroutines which regulate color, shading, and other topics in computer graphics, not only three-dimensional viewing.

4. Current Use and Future Directions

The Graphics Labs are used both with and without the supervision of an instructor. The *IRIS Programming Tutorial* [5] workbook accompanies the computer-based tools. The initial distribution of the Tutorial was in September, 1986, so feedback from the field is limited at this date. In the classroom, students are vocal about its helpfulness.

Labs for demonstrating advanced color theory are planned. The RGB (Red, Green, Blue) model will be represented by a three-dimensional color cube, with each axis weighing a different color component. The HSV (Hue, Saturation, Value) model will be displayed as a single, hexagonal cone; the HLS (Hue, Lightness, Saturation) model, seen as a double, hexagonal cone. [3] Other topics for demonstration are the ambient, diffuse and specular components for lighting a three-dimensional object.

As PHIGS [2], GKS-3D, or another 3-D graphics standard asserts itself, computer-based training tools will help instructors cope with the flood of programmers who need to understand 3-D. Since PHIGS extensively uses display list structures, a graphical description of the contents of a structure may be maintained in a view window. The Incense [6] pictorial paradigm is ideal for representing display lists, particularly the hierarchical structures of PHIGS.

5. Acknowledgments

The Graphics Labs and IRIS Programming Tutorial were developed by the efforts of the Tutorial Design Team. Amy Smith prepared the Tutorial workbook, which accompanies the Graphics Labs and edited this article. Mike Clark, Thant Tessman, Vince Uttley, and Mason Woo implemented the Graphics Labs. Zsuzsanna Molnar and Susan Luttner supervised the project, and many, many people at Silicon Graphics helped release it to IRIS users. Special thanks also to the dozens of students who tested our software during development.

6. Bibliography

[1] Brown, Marc H. and Robert Sedgewick. "A System for Algorithm Animation," *Computer Graphics (ACM) 18,* 3 (July, 1984), 177-186.

[2] Brown, Maxine D. and Michael Heck. *Understanding PHIGS,* TEMPLATE, San Diego, California, 1985.

[3] Foley, James D. and Andries van Dam. *Fundamentals of Interactive Computer Graphics.* Reading, Massachusetts: Addison-Wesley Publishing, 1982.

[4] Gurwitz, Robert F., Richard W. Thorne, and Andries van Dam. "BUMPS: A Program for Animating Projections," *Computer Graphics (ACM) 14,* 3 (July, 1980), 231-238.

[5] *IRIS Programming Tutorial,* Version 1.0., Silicon Graphics Inc., Mountain View, California, Document Number 007-1103-010.

[6] Myers, Brad A. "Incense: A System for Displaying Data Structures," *Computer Graphics (ACM) 17,* 3 (July, 1983), 115-125.

[7] Thornton, Robert W. "The Number Wheel: A Tablet Based Valuator for Interactive Three-Dimensional Positioning," *Computer Graphics (ACM) 13,* 2 (August, 1979), 102-107.

IRIS is a trademark of Silicon Graphics.

Mason Woo
2011 Stierlin Road
M/S 2L/420
Mountain View, CA 94043
(415) 962-3314

Teachers' Perceptions of Needs Related to Implementing Computer-Based Instruction

David W. Dalton

Indiana University

Abstract

This presentation describes a survey study designed to ascertain the perceptions of teachers toward the implementation of computer-based education. The results of the survey suggest that 1) teachers perceive the computer primarily as an instructional tool or a supplement to their own instructional programs; 2) word processing is the most common computer-related activity in the class rooms of these respondents; and 3) these respondents believe that the single most important priority related to implementing CBE is providing learners with basic computer literacy and keyboarding skills.

Although there is considerable and mounting evidence to suggest that computerized learning systems offer an effective and pleasant alternative to "conventional" classroom instruction, recent studies at Johns Hopkins and elsewhere suggest that the overall impact of the computer on instruction is still relatively insignificant.

A wide variety of factors have been offered to account for this lack of overall impact. Some have suggested that there may simply be too few computer workstations available to offer a meaningful alternative for teachers. Others note that there exist a number of prevalent myths about the nature of computer-based education (CBE) that cast significant doubts in the minds of would-be computer users (Hannafin, Dalton, & Hooper, in press). Still others argue that computers have yet to be tried on any large-scale, and hence, represent an unknown quantity in the minds of many educators (Bork, 1986).

Clearly, there is merit in each of these viewpoints and there are undoubtedly many other obstacles to meaningful CBE implementation. However,, this paper takes the position that the greatest single hurdle that must be overcome so that CBE can have a meaningful impact on classroom instruction remains the attitudes and perceptions of those most charged with the day-to-day instruction: the teachers.

An important first step in promoting the appropriate use of CBE is then understanding the perceptions that teachers hold about computers. The purpose of this study was to determine 1) how teachers are currently using computers to support their classroom activities, 2) how teachers wish to use computers in the future, and 3) what factors teachers believe most prevents them from more effectively using computers.

Methods

Subjects and Procedures

Sixty-eight teachers in a rural K - 12 school district in northwestern Indiana were selected for participation in this study. Because a major university is located within this community, the students are representative of a somewhat more able population than average. In general however, the concerns of most of the teachers who participated in this study are typical of teachers in general. Approximately 60% of the selected sample (n = 40) teachers returned completed surveys.

A survey instrument was developed to assess the respondents perceptions regarding the implementation of computer-based instruction. The instrument was validated using a sample of five volunteer respondents prior to its distribution. Surveys were distributed to the sample via intra-district mail and were returned to a teacher coordinator within the district.

Survey Items

The first item was an open-ended question asking respondents to describe their present computer-related instructional activities in all aspects of their teaching responsibilities.. The second paralleled the first, except that it asked respondents to describe how they wished to use computers in the future. Item 3 asked the respondents to rate the following five computer instructional applications in terms of their overall priority for implementation: 1) remediation, 2) enrichment or extension activities, 3) problem-solving simulations, 4) word processing, spreadsheets, etc., and 5) basic skills instruction. This item was accompanied by an open-ended section for additional comments.

The next item pertained to the teachers' perception of needs related to CBE within their district. Respondents rated the following problems on a scale of 1 to 5, with 5 representing "least critical" and 1 representing "most critical:" 1) lack of sufficient hardware, 2) lack of appropriate software, 3) lack of teacher training, 4) lack of information about available computer resources, 5) lack of information about integrating computers into instructional programs, and 6) lack of available resource personnel.

Item 5 used an open-ended format and asked respondents to describe their single greatest impediment to implement CBE in their own instructional circumstances. In a similar manner, the next item asked for a description of the single most important computer-related skill that learners most need to learn.

Item 7 asked respondents to state whether they preferred that computers be implemented by a single department, by a computer department, or integrated throughout all departments. Finally, respondents were asked to rate the overall support they felt was provided by the district for implementing CBE within their instructional programs.

Results

Current Use

The results for the first item are contained in Figure 1. When describing their current uses of computers, elementary teachers identified five major areas of use: 1) teacher record keeping and preparation (28%), 2) enrichment and extension activities (11%), 3) computer literacy, primarily keyboarding activities (11%), 4) basic skills instruction (22%), and 5) word processing and other applications (11%). An additional 17% of the respondents indicated no current computer use.

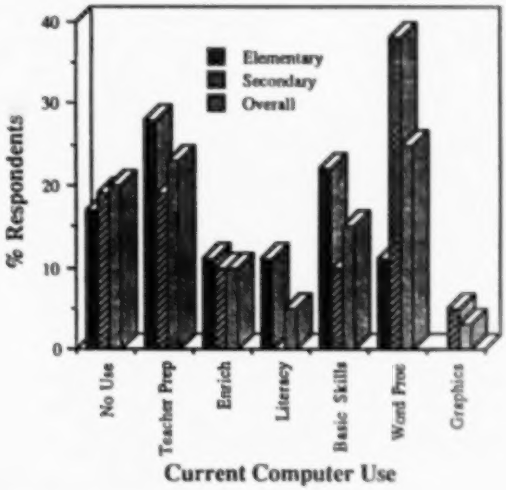


Figure 1. Current computer use.

Secondary teachers described similar applications: 1) teacher use (19%), 2) enrichment/extension (10%), 3) basic skills instruction (10%), 4) and word processing and other applications (38%). In addition, 5% of the secondary respondents indicated using graphics programs, either for the generation of graphics or for publication purposes. Nineteen percent indicated no current computer use. Overall, the results for the entire sample were: 1) teacher preparation: 23% ; 2)enrichment/extension: 10%; 3) computer literacy: 5%; basic skills instruction: 15%; 4) word processing and other applications: 25%; and 6) graphics: 3%. In total, 20% of the respondents indicated no current computer use.

Desired Uses

The results for Item 2 are contained in Figure 2. Elementary teachers expressed an interest in four major computer applications for future use: 1) enrichment and extension activities, primarily problem-solving simulations (33%), 2) computer literacy, primarily keyboarding (6%)), 3) basic skills instruction (44%), and 4) word processing and other applications (11%).

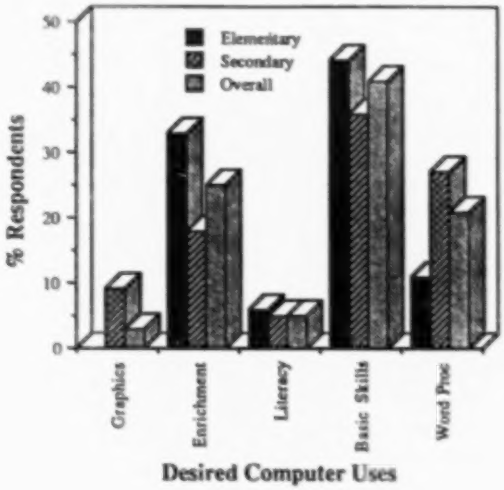


Figure 2. Desired computer uses.

Secondary teachers generally identified much the same desired uses: 1) enrichment/extension (18%), computer literacy (5%), basic skills instruction (36%), and word processing and other applications (27%). Again, graphics production and desktop publishing were also identified as potential applications (9%). Overall, 25% of the respondents wished to use the computer fro enrichment/extension, 5% for computer literacy, 41% for basic skills instruction, and 21% for word processing and applications.

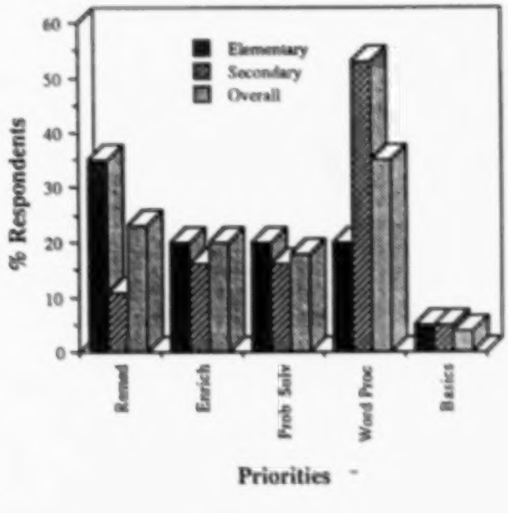


Figure 3. Computer-related priorities.

Priorities

The results from Item 3 are shown in Figure 3. When asked to rate which applications should be taught, 35% of the elementary respondents ranked remediation as most important, followed by enrichment, problem-solving and word processing (20% each), and basic skills instruction (5%).

Secondary instructors rated word processing as most important (53%), followed by enrichment and problem-solving (16% each), remediation (11%), and basic skills instruction (5%). Overall, 35% of the respondents stated that word processing was most important, followed by remediation (23%), enrichment and problem-solving (20% and 18% respectively), and basic skills.

Problems

The results from Item 4 are displayed in Figure 4. Elementary teachers rated a lack of appropriate software as the most critical problem they faced in implementing computers (30%), followed by lack of hardware and information about integrating computers into the curriculum (20% each), and lack of training and resource personnel (15%) each). Secondary teachers rated both software and training as the most pressing problems (22%), followed closely by a lack of information about local resources, and lack of resource personnel (13%) each). Overall, teachers rated lack of software as their greatest problem (26%), followed by lack of training and information about integrating computers into the curriculum (19% each), lack of hardware (16%), lack of resource personnel (14%), and lack of information about local computing resources (7%).

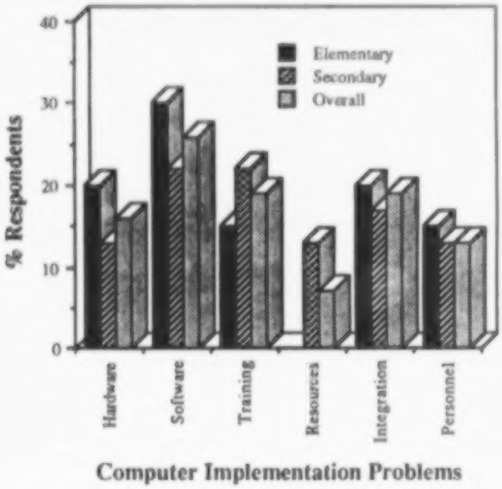


Figure 4. Computer implementation problems.

Needs

When asked to identify the most critical need related to the implementation of computers in education, elementary teachers rated a lack of software as most critical (35%), followed by insufficient release time (29%), insufficient hardware and resource personnel (18% each).

Secondary teachers identified lack of training as their most critical need (39%), followed by lack of hardware (33%), lack of software and time (11% each), and insufficient resource personnel (6%). Results for the district as a whole were: lack of hardware: 27%; lack of software and training: 22% each; insufficient time: 19%; and the lack of resource personnel (11%).

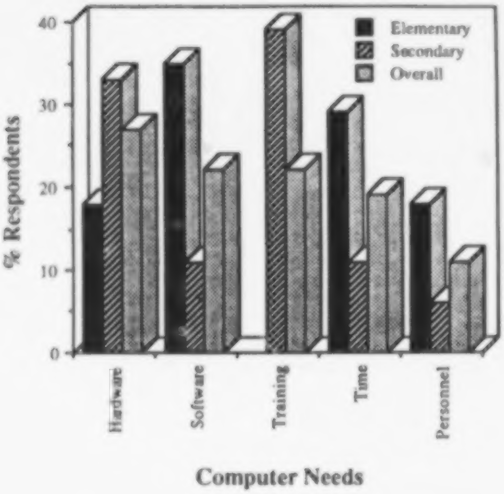


Figure 5. Critical computer-related needs

Student Skills

When asked to identify the most important skills that their students should be taught, 68% of elementary teachers rated computer literacy and keyboarding as the most important, 11% rated programming as most important, and 5% rated word processing as most important. Sixteen percent of these

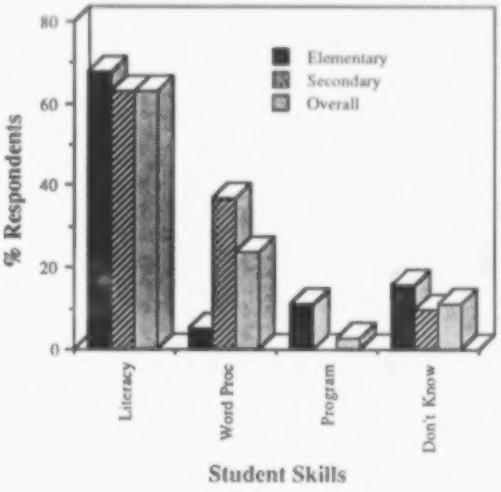


Figure 6. Most important student skills.

respondents stated that they lacked enough information to accurately state what computer skills should be taught. Secondary teachers rated computer literacy and keyboarding as most important (63%), and 37% rated word processing as most important. Overall, 63% rated computer literacy most important, 24% word processing, 3% programming, and 11% said that they did not know which skills should be taught.

Implementation

When asked how the district should implement computers, 82% of the elementary teachers stated that computers should be integrated into each curricular are and 18% stated that computers should be taught as a separate area. The secondary and overall results were very similar: 81% and 82% for integrating within the curriculum, respectively, 19% and 18% for teaching computers as a separate content area.

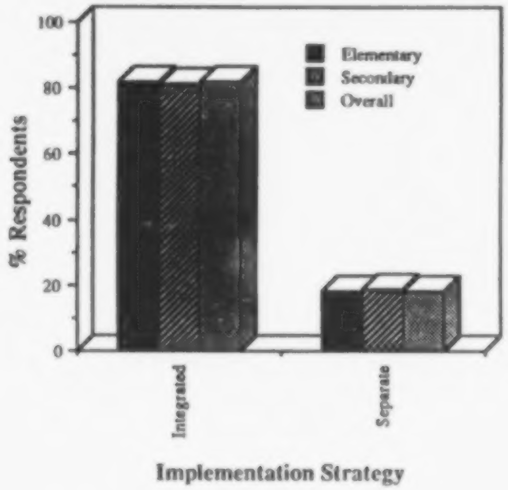


Figure 7. Computer implementation strategies.

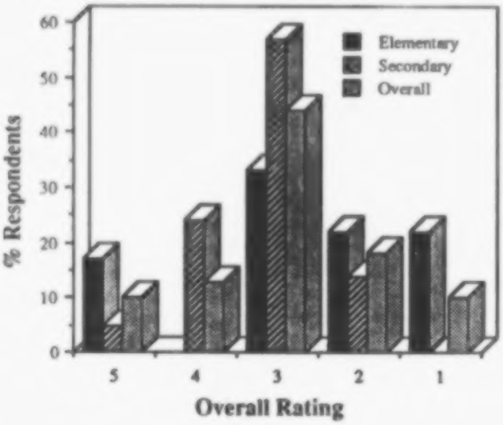


Figure 8. Overall rating of district support.

Overall Rating

When asked to rate the overall support provided by their district in implementing computers in education on a scale from 1 to 5, where 1 represented "no support" and 5 represented" excellent support," 17% of the elementary respondents rated the support as a 5, 33% rated the support as a 3, and 22% each rated the support provided as a 2 and a 1. Five percent of secondary teachers rated districts support as a 5, 24% as a 4, 57% as a 3, and 14% as a 2. Overall, 10% rated district support as a 5, 13% as a 4, 44% as a 3, 18% as a 2, and 10% as a 1.

Implications

Three major implications can be drawn from the findings of this study. First, the most common current and desired uses of the computer in instruction in this sample was word processing. Clearly, teachers in this sample clearly perceive that the computer is a powerful tool to help students develop writing skills across the curriculum.

Since teachers obviously feel quite comfortable with word processing themselves, this result may reflect a lack of information or training on these other applications. In fact, data and a number of comments from individual teachers support these conclusions. However, the most likely explanation for such a heavy emphasis on word processing may be simply a lack of appropriate software to support other types of computer activities. Ratings of problem areas tend to support that the lack of software is the single most pressing problem for most of these teachers.

Second, teachers in this sample seem to view the computer best as a tool, not as a tutor. In many cases, these teachers seemed reluctant to delegate much of their classroom teaching responsibilities. Instead, many seem to prefer implementing the computer as a supplement to their existing teaching methods, not as a replacement.

Third, the key skill area identified by nearly two-thirds of these respondents was in basic computer literacy skills. Once again, keyboarding seems to pose a major barrier to successful computer-based education. Although many teachers rated these skills of extreme importance, only 5% expressed an interest in teaching them personally. Clearly, no easy curricular solution has been found for this dilemma. Compounding the problem is the fact that teachers in this study overwhelmingly asserted that computer education should be integrated across the curriculum and not offered as a separate area. Obviously, this result begs the question, "Where will keyboarding be taught?"

Finally, although a fairly major commitment has been made in terms of financial and human resources within this school district, a sizeable number of teachers, approximately 43%, currently do not use computers with their students. Based on overall ratings of the districts perceived support, it seems clear that greater out-reach efforts will be required to encourage many elementary teachers to begin computer education.

References

Bork, A. (1986). Issues in teaching with technology: A professor's perspective. *Proceedings of the Eighth Conference of the University/Urban Schools National Task Force.*

Hannfain, M., Dalton, D., & Hooper, S. (in press). Computers in education: 10 myths and 10 needs. *Educational Technology.*

David W. Dalton
4217 Penn Court
Bloomington, IN 47401
(812) 333-9619

Computers in Education: What's Really Keeping Them Out?

Michael J. Hannafin, The Pennsylvania State University

David W. Dalton, Indiana University

Simon R. Hooper, The Pennsylvania State University

Abstract

Athough computers have presented unprecedented changes in society, they have met with considerable resistence in our schools. Critics point to a virtual absence of meaningful computer activity in the schools as indication that they have simply not responded to the needs of a changing society. This paper takes the position that much of the problem stems from beliefs regarding the impact of the computer on school systems, students, and teachers. Some of these concerns are based in reality, others are not. The purpose of this paper is to describe several of the myths that have inhibited the emergence of computers in the schools, and to identify several "real" needs for the advancement of computers in education.

Myths Concerning Computers in Education

Teachers are too busy When perceived as an additional burden on the already overtaxed teacher, computers will never realize their educational potential. Although the goal of computers in education, as in society, is to replace or simplify functions, and not simply to add responsibilities we have become so overcome with the burdensome task of teaching that we have neither recognized nor utilized technology that offers legitimate potential for improvement. We have not invested the time necessary to save time.

Computers are "fads": Here today, gone tommorrow. Many have argued that the computer is simply the latest in a series of instructional fads that will fade gradually into obscurity. However, computers have permeated virtually all aspects of society. Since schools mirror the priorities of their respective societies, and the image is clearly one of investment in computer technology, we will undoubtedly experience increased pressures to orient students to our fast paced, technologically-oriented society.

Computers are too expensive, but more expensive than what? They are, perhaps, typically more expensive than current textbooks, workbooks, chalkboards, and other print-based media. When the formidable start-up, maintenance, and software costs are identified, many consider the cost as prohibitive compared with conventional print-base materials. However, the additional costs, while formidable, are largely one-time start-up costs that, when cost-factored over many years of formal schooling, may ultimately be less expensive than other alternatives.

Computers provide impersonal learning environments. Many argue that computers create a sterile, impersonal climate where little, if any, nurturing occurs. This position appears to have evolved from the belief that the "human touch" of the teacher helps to engender values about human relationships. However, many of the presumed affective qualities of teachers are often inconsistently and ineffectively applied. In contrast, computers have the potential to emulate the best of methods in student interaction and although the individualized learning potential of the computer is impressive, it does not imply that all computer-based instruction requires the physical isolation of students.

There are not enough computers available in the schools. Recently, a study conducted at Johns Hopkins University suggested that the number of hardware systems must be increased dramatically if microcomputers are to have any real impact in education. Yet, truly creative and talented classroom teachers may find extraordinary uses for a classroom full of microcomputers or even a single computer. For the unmotivated or resistent teacher, however, one computer is too many and one thousand is too few.

Inadequate supply of well-designed software. During recent years, software quality and quantity have improved noticeably. Several software production companies now market exceptional instructional games and high interest simulations, with sufficient scope to permit on-going use through a series of related activities. In addition, textbook publishers have demonstrated heightened interest since, with the availability of significant numbers of computers, large-scale commitment to curriculum-based instructional software has become potentially profitable.

Computers are only useful for teaching basic, low-level skills. The computer makes available teaching process options that are simply unavailable through other methods. The teaching and practice of creative writing and composition skills, the simulation of scientific events impossible to observe with the naked eye, the value-free compassion of the computer as counselor, and the application of the computer as an expert problem solver and diagnostician are but a few of the enormously powerful applications that transcend simple low-level learning. We may relegate the computer to only a portion of its true potential, but this is a problem of human perception rather than technologic capability.

Insufficient opportunities for teacher training. To implement meaningful computer-based education, meaningful preparation is needed. The issue is not simply one of opportunity but one of emphasis. Although most teacher training programs and individual school disaricts provide some form of computer preparation to teachers, many of the courses offered to teachers emphasize low-level, "literacy" skills so that teachers become aware of the computer, but fail to receive adequate preparation in its integrated uses in education. The opportunity is provided, but the emphasis is mis-directed.

Computers will replace teachers. Perhaps this myth, more than others, has created subconscious resistance to the installation of computers in most schools. This sentiment, however, presents a significant barrier to the infusion of computers in education. The computer is not so much a threat but a liberating device for teachers. Certainly, the computer will effect the availability of teaching positions in the future, but it should also improve the quality and job satisfaction of the teaching profession . The computer can free the teacher to concentrate on the needs of individuals, the climate and structure of the classroom, and activities that nurture and promote a wide variety of educational goals.

Needs Concerning Computers in Education

Comprehensive curriculum featuring computer-based activities and methods. There still exists virtually no comprehensive computer-driven curriculum. Several courses focusing on the computer per se have been developed, but few that offer a significant "computer as teacher" option for specific non-computer subject areas. These comprehensive computer curricula will almost certainly be forthcoming in the near future, but the sooner such systems are made available the more rapidly role shifts can be studied and understood.

"Friendlier" computers. As computers have become more powerful and miniaturized, many have also become easier to learn and use (Wagschall, 1984). Computers must continue to become easier to operate, and more transparent to the user. Several steps have been made in this direction, including advances in easy-to-use authoring systems, "pull down" menus, and integrated support. Advances are needed to simplify further the use of simple utility programs such as grade books, and the methods for establishing "dialog" between the computer and the student or teacher. Educators must gain routine, automatic comfort with computer use, analagous to the almost unconscious steps involved in driving a car.

Cost-sharing arrangements. Central media centers need to be established to showcase and lend computer hardware and software. In addition, the concept of "economy of scale" should be applied to computer-based education. It makes little sense for individual teachers or schools reinvent the wheel and develop their own computer materials. State and national clearinghouses need to address problems associated with cost, classroom integration, computer accessibility, and equity.

Emphasis networking. The prohibitive costs of "one per station" software, and hardware have accelerated interest in computer networks. We must develop networks that permit the sharing of expensive hardware and software and permit access to significant software banks or curricula, such as possible using hard disc systems. It seems unlikely that significant computer growth can occur without such systems.

Teacher enthusiasm equal to student enthusiasm. The fascination of children with computers has grown dramatically. Individual teachers, as well as the teaching profession collectively, must invest in the potential and promise of computers if efforts are to succeed. Indeed, teachers must invest more effort and attention to computers in order to ensure advancement of their students.

Emphasis on supporting the role of the teacher. Many innovations fail to gain acceptance in the schools because they simply add to the load of already an overburdened teacher. There is no need to add computers, for the most part, to curriculum: instead, infuse computers into normal routines, capitalize on the unique aspects of computers to motivate and support instruction, and use them to reduce or eliminate the boring, non-creative chores of teaching.

A committment to improvement through change. The commitment to make change effective and productiveis essential to the installation of any innovation. The educational system has become notorious for being impervious to improvement through change. Clearly, an institutionalized or mandated commitment to change must be made at all levels.

Teacher training emphasizing integrated computer uses. Substantially greater effort must be made to improve the quality, and nature, of teacher training efforts. Better methods for integrating the computer in the classroom must now be explored and instituted within teacher training institutions and school districts.

More and better research. Research in the design of CBI has yielded important insights into the characteristics of effective software. Still, the absence of research relevant to the implementation of computer-based education is seriously lacking. Intensive efforts to study the effective use of technology in schools and extend such methods to other schools is urgently needed.

Emphasis on computer experiences and opportunities for disadvantaged and underrepresented learners. Computer exposure and fluency have evolved largely along socioeconomic lines. One consequence of this inequity has been to widen further the base of experiences available to privileged versus underprivileged students. Children from economically disadvantaged backgrounds have been "locked out" of many computer experiences A similar concern must be expressed for gender inequities that appear to have surfaced related to computer careers for young women. There is evidence that high school girls tend to perceive the computer, like mathematics and science, as domains reserved for males. This situation is troublesome and must be addressed.

The Effects of Individualistic Versus Cooperative Computer-Assisted Instruction on Student Performance and Attitudes

David W. Dalton, Indiana University

Michael J. Hannafin, The Pennsylvania State University

Abstract

In this study, the performance and attitudes towards instruction of learners working individually was compared with learners working cooperatively in two-member teams. A total of 60 eighth graders received either an individual or a cooperative CAI lesson on the human reproductive system. Students in the cooperative group scored significantly higher on an achievement measure. An interaction was detected between instructional method and gender for attitude, with males responding most favorably to individual, and females most favorably to cooperative, CAI instructional methods.

During the past decade, computer-based instruction has been effective in improving both learner performance and attitudes towards instruction (see for example, Kulik, Kulik, & Bangert-Drowns, 1985). Since much of the assumed benefit of CAI has centered on individualistic instruction, much of the CAI research has focused on one-to-one teaching methods. However, isolation of learners from peers during these one-to-one lessons may be undesirable in many instances (Johnson, et al, 1981; Moskowitz, et al, 1983; Slavin, 1980; Webb, 1982b). Yet, some theorists have suggested that cooperative learning effects are not uniform across learners of different abilities, and therefore, appear to be influenced significantly by several individual learner variables (Webb, 1982a).

Recently, grouping effects have been studied for computer-based learning (Dickson & Vereen, 1983; Johnson, Johnson, & Stanne, 1986; Webb, 1984; Webb, Ender, & Lewis, 1986). Along with providing the potential to overcome the logistical hurdles created by computer shortages in schools, cooperative CAI has proven effective in the development of social skills, creative thinking, and overall academic performance (Johnson & Johnson, 1986). In short, it may be not only unnecessary to implement many CAI lessons one-on-one, but unwise as well.

Many questions as to the effects of individualistic versus cooperative computer-based learning remain. The purpose of this study was to examine the effects of individualistic versus cooperative methods of CAI on both learning and attitudes.

Methods

Subjects

The subjects were 60 eighth graders selected from two sections of a health class. The sample consisted of approximately equal numbers of males and females, and comprised roughly typical urban-suburban ethnic representation.

Materials

The basic instruction for this study consisted of a 30 minute lesson on the anatomy and physiology of the human male and female reproductive and urinary systems. Each lesson segment began with a tutorial describing the size, shape and location of the organ, followed by a computer-generated graphic depicting the organ in the body. A brief explanation of the organ's function was then given, again in tutorial form. Upon completion of each segment, the learners were asked to recall the name and function of the organ. Two lesson variations were developed: individualistic and cooperative.

Individualistic instruction. In this treatment, learners were assigned individually to a computer terminal and told to work independently until the entire lesson had been completed. Interactions with other learners were discouraged and the teacher acted only to answer questions about the operations of the computer system.

Cooperative instruction. In this treatment, each learner was randomly assigned a fellow classmate as a partner with whom to complete the lesson. These learner pairs were then instructed to begin the lesson and to "work together as much as possible" during the actual lesson and to take turns at the keyboard entering the partnership's responses. After the first lesson segment concerning the male anatomy was completed, the lesson prompted the partners to switch roles at the keyboard and once again reminded them to "work together" on the lesson material. In this manner, each member of the team assumed role of "communicator" and was required to enter the team's consensus responses during the lesson.

Dependent Measures

Posttest. The posttest consisted of a total of 48 print-based items, 36 multiple choice items on the names and functions of the human reproductive system organs and 12 items requiring learners to label line drawings of the reproductive systems. Split-half reliability of the overall test was found to be .89 using data obtained during the study.

Attitude survey. Learners rated anonymously a 20-item print-based Likert-type survey. This survey contained items designed to assess attitudes toward the instruction and content.

Split-half reliability of the attitude survey was .86.

Procedures
Students were randomly assigned to their repective treatments where they completed the instruction. All students were given a general explanation as to the nature of the lesson, as well as to the procedures needed to operate the CAI lesson. Specific directions for implementing cooperative versus individualistic strategies were provided to respective groups by the teacher. In addition, learners in cooperative group received a reminder to "work together" at the lessons midpoint. Students then received the instruction in accordance with their group assignment. In order to avoid interference and contamination across treatments due to the inherent differences in the treatments, only one lesson version was implemented at-a-time. Immediately following the lesson, students were given the posttest and the attitude survey.

Experimental Design Data Analysis
This study employed a completely crossed 2x2x2 randomized block factorial design, featuring two levels of instruction (cooperative, individualistic), gender (male, female), and ability (HI, LO), split at the median of the sample's Comprehensive Test of Basic Skills total scores. Posttest scores were analyzed through fixed-effects ANOVA procedures. Comprehensive Test of Basic Skills (CTBS) total scores served as the ability blocking variable. Attitude data from this study were analyzed with fixed-effects ANOVA procedures.

Results

The results for the posttest are contained in Table 1. The means of the individualistic (53.10%) and the cooperative (70.77%) instructional groups were significantly different, $\underline{F}(1,54) = 9.95$, $\underline{p} < .005$.

Table 1. Mean percent scores and standard deviations for the posttest

			Instructional	Group	
Gender	Ability		Individual	Cooperative	Total
Male	HI	Mean	62.86	79.14	71.00
		SD	(18.46)	(19.08)	(19.92)
		n	7	7	14
	LO	Mean	42.00	58.50	50.80
		SD	(23.17)	(9.73)	(18.71)
		n	7	8	15
Female	HI	Mean	71.57	82.50	77.40
		SD	(15.01)	(15.34)	(15.70)
		n	7	8	15
	LO	Mean	38.13	64.00	51.06
		SD	(17.89)	(8.86)	(19.09)
		n	8	8	16
Total	HI	Mean	67.21	80.93	74.31
		SD	(16.79)	(16.64)	(17.83)
		n	14	15	29
	LO	Mean	39.93	61.25	50.94
		SD	(19.85)	(9.43)	(18.59)
		n	15	16	31

The cooperative group yielded consistently superior performance across both gender and ability. As expected, HI ability students out-performed LO ability students (74.31% vs. 50.94%), $\underline{F}(1,54) = 18.90$, $\underline{p} < .001$. No significant differences were found between the males and females, and no interactions were detected for posttest performance.

The results for the attitude survey are contained in Table 2. Though main effect differences were not detected, there was a significant interaction between instructional group and gender, $\underline{F}(1,53) = 5.26$, $\underline{p} < .05$.

Table 2. Mean percent scores and standard deviations for the attitude survey.

			Instructional	Group	
Gender	Ability		Individual	Cooperative	Total
Male	HI	Mean	46.57	50.57	48.57
		SD	(13.73)	(9.38)	(11.49)
	LO	Mean	55.43	39.63	47.00
		SD	(14.72)	(17.07)	(17.47)
Female	HI	Mean	58.00	56.13	57.00
		SD	(16.18)	(6.36)	(11.55)
	LO	Mean	36.88	61.00	48.94
		SD	(18.67)	(11.29)	(19.43)
Total	HI	Mean	52.29	53.33	52.93
		SD	(15.59)	(8.13)	(12.10)
	LO	Mean	45.53	50.31	48.00
		SD	(18.95)	(17.81)	(18.22)

Notes: n's per cell and total N shown in Table 1.

This interaction, shown in Figure 1, was characterized by males responding most favorably after individual rather than cooperative methods (51.00 vs. 44.73), while females responded most favorably following cooperative versus individual instruction (46.73 vs. 58.56).

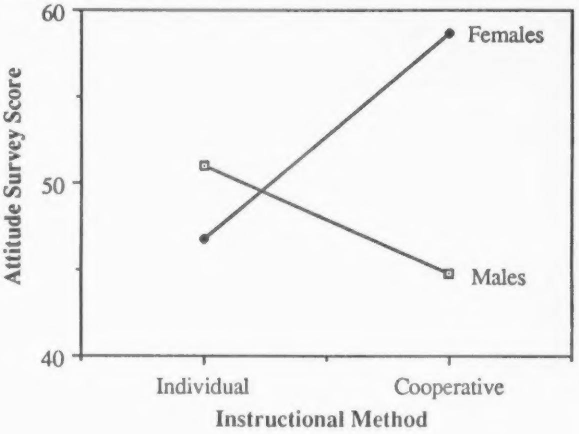


Figure 1. Interaction between instructional method and gender.

Discussion

The finding that cooperative methods yielded better achievement than individual methods is consistent with research featuring both traditional and CAI teaching methods. One of the criticisms of cooperative methods has been the inability to assure the kind of sustained individually customized engagement available through individualistic CAI lessons. This potential limitation often renders group methods ineffective, permitting individuals within the group to dominate lesson activities while others receive limited interaction. How, then, can cooperative methods sustain attention and engagement for groups of students working simultaneously?

In this study the cooperative groupings involved only two students, permitting (and perhaps even requiring) continuous involvement by both students throughout the lesson. The group size, paired with the consensus format for interaction and the requirement to rotate the role as communicator, forced a higher and more sustained level of engagement than often possible with traditional one learner per machine CAI interactions. In addition, the interaction fostered through cooperative encounters may have served a number of organizing and clarifying functions, with sudents receiving perspectives from partners that enabled each partner to confirm, correct, or clarify comprehension of the lesson concepts.

What of the apparent loss of true individualization created by the use of consensus versus individual responses during CAI? Can truly relevant and individualized instruction be provided when responses are not the result of individual and unique student processing? It seems certain that individualization of responses has been retained, but is applied to consensus perceptions rather than individual perceptions. Cooperative team members retain the rationale for individual perspectives whether or not this perspective is subsequently retained as a consensus response. The consequences of responding, as a result, can be evaluated individually from both unique (per the isolated opinion of each student) and collective (per the consensus opinion of the cooperative group) perspectives. In effect, the cooperative process, when properly implemented, does not limit the value of CAI interaction, but instead enriches it by expanding the perspectives of individuals to a range of alternatives and opinions.

The effects of different instructional methods on student attitude is also of interest. The present findings suggest that cooperative CAI is more appealing to females than to males. Though gender differences have been identified previously for cooperative methods (Webb, 1982a), the effect detected in this study may have been influenced largely by social and instructional content factors. The content selected for the study, the human reproductive system, is more inherently sensitive to gender-differentiated reactions than neutral subject matter -- especially for eighth graders. The subject matter proved sensitive for both males and females: Males rated individual methods more favorably than cooperative methods, but females preferred cooperative methods to an even greater degree than males preferred individual methods. An interesting follow-up question concerns the effects of like-versus-mixed gender pairings on ratings to sensitive instructional content. It seems likely that ratings to the sensitive lesson content used in this study will be moderated further by the potential for embarrassment among mixed gender pairs. Cooperative group pairings were randomly assigned in this study, thereby precluding examination of the effects of male-male, female-female, and mixed gender pairings on resulting attitudes.

Interestingly, instructional methods effects for attitude were negligible, even though academic performance was clearly superior in the cooperative learning groups. Typically, attitudes correlate positively with perceptions of success during lessons: increases in academic success are usually accompanied by improvements in attitude, while decreases in success are associated with decrements in attitude. This, however, was not the case in the present study. Within the limitations associated with sensitive lesson content, social dynamics may override the affective impact typically associated with lesson success.

The utility of cooperative CAI paradigms has important practical implications. The limited availability of computers in schools mandates that either access be systematically restricted, or that alternative methods for using the computer be developed. Findings from the present study, as well as a growing number of additional studies, suggests that cooperative CAI methods often provide superior performance compared with individual notions of CAI across academic as well as social and psychological domains. Cooperative methods may provide not only alternative, but often preferred, CAI methods.

References

Dickson, W.P., & Vareen, M.A. (1983). Two students at one computer. *Theory Into Practice, 22*, 296-300.

Johnson, D.W., & Johnson, R. (1986). Computer-assisted cooperative learning. *Educational Technology, 26*, 12-18.

Johnson, R.T., Johnson, D.W., & Stanne, M.B. (1986). Comparison of computer-assisted cooperative, competitive, and individualistic learning. *American Educational Research Journal, 23*, 382-392.

Johnson, D.W., Maruyama, G., Johnson, R., Nelson, D., & Skon, L. (1981). Effects of cooperative, competitive, and individualistic goal structures on achievement: A meta-analysis. *Psychological Bulletin, 89*, 47-62.

Kulik, C., Kulik, J., & Bangert-Drowns, R. (1985). The importance of outcomes: A reply to Clark. *Journal of Educational Computing Research, 4(2)*, 381-87.

Moskowitz, J.M., Malvin, J.H., Schaeffer, G.A., & Schaps, E. (1983). Evaluation of a cooperative learning strategy. *American Educational Research Journal, 20*, 687-696.

Slavin, R.E. (1980). Cooperative learning. *Review of Educational Research, 50*, 315-342.

Webb, N.M. (1982a). Group composition, group interaction, and achievement in cooperative small groups. *Journal of Educational Psychology, 74*, 475-484.

Webb, N.M. (1984). Microcomputer learning in small groups: Cognitive requirements and group processes. *Journal of Educational Psychology, 76*, 1076-1088.

Webb, N.M. (1982b). Peer interaction and learning in cooperative small groups. *Journal of Educational Psychology, 74*, 642-655.

Webb, N.M., Ender, P., & Lewis, S. (1986). Problem-solving strategies and group processes in small groups learning computer programming. *American Educational Research Journal, 23*, 243-261.

USING COMPUTER SIMULATED BASE TEN BLOCKS TO TEACH DOUBLE-DIGIT ADDITION AND SUBTRACTION

Gary A. Deatsman - The Ohio State University at Newark

ABSTRACT

Base ten blocks are manipulative devices often used to teach arithmetic. This paper describes the design, creation, and testing of computer programs to teach double-digit addition and subtraction with regrouping (carrying and borrowing) using animated graphic representations of base ten blocks. Informal observation of children's reactions and learning suggests that this may be an effective technique.

INTRODUCTION AND RATIONALE

People of normal intelligence who have trouble with mathematics often lack a "feel" for numbers. To them, mathematics is a meaningless jumble of rules for manipulating symbols. They lack a sufficient concept of the quantities these symbols represent. They not only become confused performing computational algorithms, but are also often unable to apply these algorithms to solve problems.

Mathematics educators have sought to prevent such difficulties by using manipulative devices such as counting beads or blocks, rods, abacuses, or "base ten blocks" which are blocks fastened into groups of ten, one hundred, etc. Children can manipulate these objects to perform arithmetic calculations and gradually proceed to abstract symbolic calculation by learning to record the results of the concrete manipulation.

Children vary greatly in their need for such activities. Some pick up the salient concepts very quickly and are soon bored if required to continue working at the concrete level when they are ready for more abstract work. Others need to do much manipulation before going on to symbolic calculation. Individualized instruction is essential, but often difficult to provide. Perhaps CAI could help fill this need, using computer animated graphic representations of manipulative devices. The programs written by the author use such representations of base ten blocks to model algorithms for adding and subtracting double-digit whole numbers.

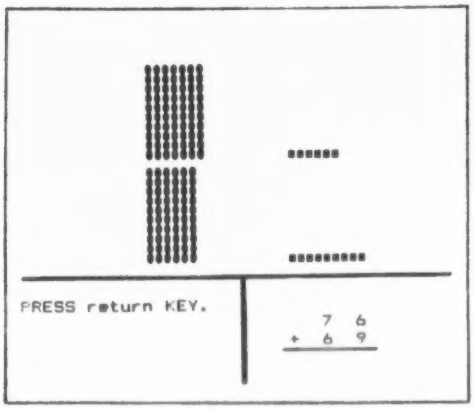
EARLY DESIGN DECISIONS

Consideration was given to allowing the children to manipulate the blocks by providing some kind of "pick up", "move", and "put down" capabilities. However, most school computers in our area are not equipped with a light pen or mouse, and these manipulations would be very cumbersome to perform with just keystrokes. Consequently, it was decided that the computer would manipulate the blocks, pausing after each manipulation to prompt for digits recording the result of the manipulation. This is called "semiconcrete animated" instruction by White and Berlin (1986). It fits into the center of their proposed two-dimensional instructional model.

The programs were written for the Apple II computer because it is so widely used in elementary school classrooms. Pascal was chosen as the programming language because it lends itself so well to writing well structured modular code. This was considered essential since many revisions and modifications were anticipated.

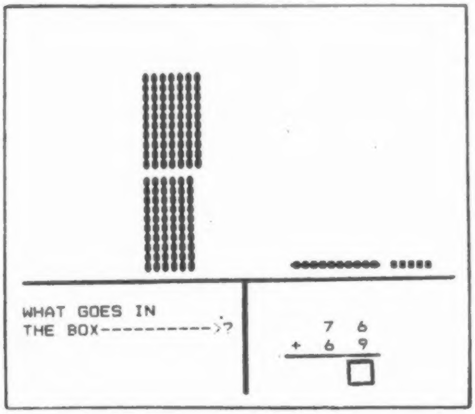
Extensive experimentation was done to find the clearest graphical representations for ones, tens, and hundreds. Some early attempts produced strange looking displays due to Gestalt phenomena. The final choice was to simply use square blocks for ones and group them into tens by drawing a line joining a row or column of blocks. One hundred was shown as an array of ten tens joined both vertically and horizontally through each block. The blocks were displayed and moved about in approximately the top two-thirds of the screen. The bottom third of the screen was used to display prompts and to receive input and to display the symbolic algorithm.

The following illustrations show a sequence of screen displays used in adding 76 and 69.



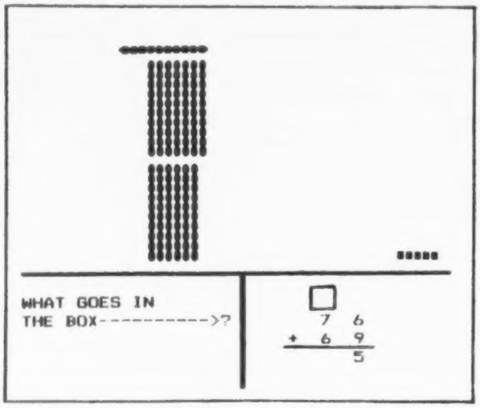
```
PRESS return KEY.

   7 6
 + 6 9
 -----
```

1. Original Display



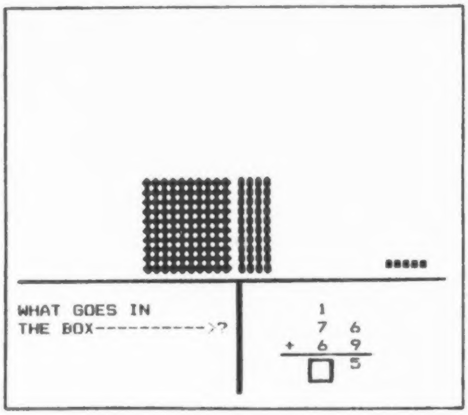
```
WHAT GOES IN
THE BOX---------->?

   7 6
 + 6 9
 -----
     □
```

2. After Combining Ones and Grouping



```
WHAT GOES IN
THE BOX---------->?

   □
   7 6
 + 6 9
 -----
     5
```

3. After Carrying from Ones



```
WHAT GOES IN
THE BOX---------->?

   1
   7 6
 + 6 9
 -----
   □ 5
```

4. After Combining Tens and Grouping

DESCRIPTION OF THE ORIGINAL PROGRAM

The first program written taught addition. It was driven by an eight choice menu. The eighth choice was to quit. The seventh choice worked problems entered by the user. The first six choices corresponded to levels of randomly selected practice problems. Practice levels 1 and 2 presented double-digit addition problems with no carries. Levels 3 and 4 involved problems with carries from the one's place. Levels 5 and 6 required carries from the ten's place and sometimes from the one's place. Levels 1, 3, and 5 prompted for a digit AFTER each graphic manipulation. Levels 2, 4, and 6 prompted for a digit BEFORE each manipulation in order to make the student anticipate the results. Optional examples were available at the start of each practice level.

Each practice session consisted of ten randomly selected problems. A motivational fantasy was included to keep students informed of their progress in an interesting way. In the fantasy a robot has been robbed of his arms, legs, and head by an evil wizard. In order to regain them he must climb ten steps in a magic staircase. Each problem done with no errors boosts the robot up one more step.

Since some children in early trials seemed to really enjoy watching him climb the steps, each error-free problem done was rewarded by the robot climbing from the bottom of the steps one more step than before. If the child made an error, various hints were given to help complete the problem at hand, but then the program returned to the menu and the practice session had to be restarted from the beginning. When ten consecutive error-free problems were done, the robot would climb to the tenth step. His arms and legs were restored, and he would thank the child for helping him.

FIRST CLASSROOM TESTING

The program was first tested in a second grade class in Autumn, 1986. The class consisted of 24 children in a rural school. The author visited the class, instructed each child in the use of the program and observed their reactions. The children were familiar with the Apple II, and needed very little instruction. Most could read the prompts easily, but a few needed some reading help at first.

The children had already been instructed in double-digit addition WITHOUT carrying, so they had very little difficulty with practice level 1. All but five completed ten consecutive error-free problems on their very first try. These five children all had trivial difficulties such as getting ahead of prompts (anticipating the NEXT answer), or inadvertently pressing a wrong digit key and not noticing it until after pressing "return". Two of them made errors near the start of the session and then went on to do ten error-free problems. The other three completed 7-9 problems before making an error and completed ten at the next session.

Two children tried level 3 before any class instruction in carrying. They received no instruction other than watching the examples available at the start of the session. They were at the computer together and watched each other's work. The first child to try problems kept putting the sum of the tens digits into the carry position, and did not realize she was making a mistake. She was confused when she was repeatedly sent back to the menu to start over. After her error was pointed out she proceeded to do ten consecutive error-free problems and complete the level. The second child (having watched the first) completed the level on the first try. This demonstrated that some children could learn carrying just by working with the program. However, the teacher felt that there was a danger that some children would experience considerable frustration if they were not first instructed in the usual way, so no further use of level 3 was made until after class instruction in carrying. While carrying was taught in the traditional way the children continued to practice with levels 1 and 2 until Christmas vacation.

Some of the children's reactions to the program were reassuring. They had little trouble reading prompts or finding the related parts of the block display and symbolic algorithm. They had no trouble counting single blocks or groups of ten. In fact, some children were very careful about checking their addition facts by counting even when they knew the answers without counting. Some other children whose knowledge of addition facts was weak counted blocks as a "crutch". Others barely glanced at the blocks once they were familiar with the symbolic algorithm. A few children seemed a bit bored by the program, but most seemed interested and some were very enthusiastic. Some showed great pride in error-free performance and worked very carefully and methodically to achieve it.

REVISIONS

Some other reactions led to extensive revisions of the program. It had deliberately avoided telling children when they had made an error, and simply gone on to help them complete the problem at hand before issuing a statement that an error had been made and returning to the menu. As noted above, this had caused some confusion to one of the children who tried carrying before class instruction. The program was changed to give an error indication immediately when a response was wrong.

The same child was confused and frustrated by being sent back to the menu after each error. This had not caused a problem in levels 1 and 2 since so few errors were made there. However, since more errors were to be expected in the higher levels involving carrying, the program was made to simply not move the robot after completion of a problem with an error. When another problem was done without error, the robot would move up another step. Thus he eventually arrived at the top step and received his stolen limbs if enough error-free problems were done, but they did not have to be CONSECUTIVE problems. A count was kept of the total number of errors made in the session. If the robot reached the top step with no errors made, he not only was given his limbs, but then his face appeared enlarged on the screen, a congratulatory message was displayed, and a siren sounded to attract the attention of the teacher.

Although some of the children enjoyed watching the robot climb from the bottom of the stairs after each error-free problem, several expressed boredom with this and asked if he could not simply climb one more step starting at his previous position. This change was made. Also, since some students tended to be confused by the difference between odd and even numbered practice levels, the number of practice levels was reduced to three: Level 1 involved no carries, level 2 required carries from the one's place, and level 3 required carries from the ten's place and sometimes from the one's place. All levels prompted for a digit AFTER each graphic manipulation.

The teacher expressed concern over the amount of time required for a child to complete a session. Some took as much as twenty minutes, and made it impossible for every child to get a turn each day at the class's one computer. The number of problems in a practice session (and hence the number of steps for the robot to climb) was reduced from ten to five. This together with the climbing of just one step at a time reduced the time for a session to between five and ten minutes.

All of the above revisions were made during Christmas break, and also a similar program was written to teach subtraction with borrowing.

FINAL TESTING

After Christmas break, when the children had received class instruction in carrying, they practiced with the new level 2 addition (carries from ones). After a few days of practice, all but seven children were observed to complete the level with no errors (and were rewarded with the siren). Five children made just one error (due to carelessness, not lack of understanding). One child made two errors, and one made three.

At the next observation session, all children observed (all but three) completed level 1 subtraction (no borrows) with no errors. Although she had not yet been taught borrowing, one child was given a brief introduction, watched the examples at the start of level 2 subtraction (borrows from tens), and then completed the level with no errors.

After the class had been instructed in borrowing, they practiced with level 2 subtraction for a few days and were observed once more. Eleven children were observed. Nine completed a level 2 subtraction session with no errors. One made one error and one appeared confused and made three errors.

EVALUATION AND FUTURE DIRECTIONS

Most children seemed to have no trouble learning to use the programs. After a few repetitions they could complete any of the practice sessions with no or very few errors. Most of them were very attentive to the graphic displays and could easily associate graphic manipulations with corresponding symbolic operations.

Of greatest importance, however, THE TEACHER FELT THAT BY USING THE PROGRAMS, THE CHILDREN WERE LEARNING CARRYING AND BORROWING BETTER THAN THEY WOULD HAVE WITHOUT THEM. She noted that in particular, some children were having difficulty understanding borrowing before using the programs. After they started practicing with the programs they soon "caught on".

Controlled testing should be done to determine the effects of the programs on learning, retention, and applicability of learning. Studies must be made to determine how best to integrate the programs into a sequence of learning experiences progressing from concrete to abstract. Other programs to simulate manipulative devices (e.g. an abacus) are planned.

ACKNOWLEDGMENTS

The author received many suggestions and comments helpful in designing the programs from professors Cary Rader, Donna Berlin, John Skowronski, and others. Principal Roger Abshire graciously allowed use of the facilities of Newton School, Newark, Ohio. Susan Keough patiently bore the

inevitable disruption of her second grade class. She and her students tested, suggested revisions, and retested the programs. It could not have been completed without them.

REFERENCE

White, A.L. & Berlin, D.F. (1986). A Model for Research, Design, and Selection of Instructional Uses of Technology in Mathematics and Science Education. 27th ADCIS Conference Proceedings, 234-238.

GARY A. DEATSMAN
RD. 3, STATE RT. 97
BELLVILLE, OH 44813
(614) 366-3321

EFFECTS OF UNIT TEST ITEM FEEDBACK ON FINAL EXAM SCORES FOR COMPUTER ADMINISTERED EXAMS

By D. L. Jelden, Ed.D.

University of Northern Colorado
Greeley, Colorado

Introduction

The educational value of test feedback has been looked at from many different perspectives. With the advent of the computer, various forms and types of test feedback could be examined in an experimental manner and the identification of the most effective type could be determined. This paper is a report of a staff study which examined several forms of unit test feedback in computer assited learning (CAL) and their effectiveness in contributing to course performance.

Nature of Test Feedback

A review of the research literature on test administration and test feedback identified several studies pertinent to this research problem. A variety of feedback modes or corrective feedback options have been given to students in a wide variety of courses and grade levels. The net result has been found to be basically the same in all cases. Item feedback during a computer administered test does improve the performance of students on the test.

A summary of the research in the field of test administration and modes of item presentation for computer managed testing offers the following conclusions:

(1) Corrective feedback does improve the achievement level of students.(Arkin, 1983), (Bender, 1984), (Hail, 1983)

(2) Retention of course meterials is enhanced with the use of corrective feedback on test items. (Hail, 1984), (Peek, 1985)

(3) Corrective feedback to students on tests should provide the correct answer and the reasons why any given item was missed. (Roberts, 1985), (Jelden, 1986), (Vitulli, 1983)

(4) Computer generated exams appear to be most beneficial to students when students can determine when to be tested and over what objectives.(Arkin, 1983), (Hagerty, 1970)

(5) Students seem to have a positive attitude toward CMI systems when used for test administration. (Gallagher, 1970), (Lawler, 1970)

(6) Immediate feedback is better than delayed feedback if retention and learning achievement is to be enhanced.(Schroth, 1984), (Plake, 1981)

(7) Low ability students seem to benefit more than high ability students from corrective test feedback.(Homa, 1984), (Bender, 1984)

(8) Students do better in course achievement when they know the status of their progress in a course or on a test than those who receive no feedback or course progress information.(Arkin, 1983)

Methods and Procedures

The data for the study were obtained from the 696 students enrolled at The University of

82

Northern Colorado in the Oceanography 110 course over several quarters. Fall Quarter 1986 was established as the control group for the study. The Winter and Spring Quarters of 1986 were established as the experimental groups which would have the feedback limitations and modes of presentation imposed on their unit tests.

Because the sample of the study was very large, (N=696), it was anticipated that any variation in treatment within any group would be rather easy to identify. Also, because each of the three groups had ACT scores on file required for admission to UNC, it was possible to compare differences between the ability levels of the groups to see if significant variation in outcome could be related to ability level. To establish the homogeneity of the population, the composite scores on the ACT test for each of the groups were obtained and mean values established and compared. This process showed the three groups similar in makeup at the .01 level.

Findings of the Study

Based on the data obtained from the students participating in the study, the following information is presented.

The first major hypothesis to be tested with the study was:

Ho1 - Students who are presented unit tests items in sequential order by objective will score significantly higher than students who are presented the unit test items in random objective order.

Based on an analysis of data obtained in the study, this was found to be true. Differences were not great in terms of point values, but it was significant at the .05 level.

The second hypothesis of the study related to the degree of test feedback and its usefulness of final exam scores. The hypothesis stated was:

Ho2 - Students who are given feedback on unit tests items of 'correct' and 'incorrect' as well as the right answer will score significantly higher on final exam than students who are provided no feedback on unit tests.

Data obtained in the study held this hypothesis to be true. Again, the differences were not great in actual numbers, (2 percentage points), but it was statistically significant at the .05 level.

In comparing the three groups relative to feedback and mode of presentation (sequential vs. random objective), final exam scores were highest when sequential presentations by objective and full feedback was used. An analysis of co-variance indicated that the feedback was more valuable than the mode of presentation although both contributed to imporved final exam scores. The adjusted means in the analysis of co-variance for the three groups with the final score being the dependent variable indicated that the feedback on individual test items was more helpful than no feedback and that sequential objective presentations were more helpful than random objective presentations.

To summarize the general results of the findings of this section, a statement of the hypothesis tested was as follows:

Ho3 - Students who are administered unit test items in sequence by objective and given feedback of 'correct' and 'incorrect' as well as the right answer will score significantly higher on final exams than students given similar information on unit test items but administered in random objective order.

Data obtained from the study did prove this hypothesis to be true. As with the previous findings, the differences were numerically small but they were significant at the .05 level.

The last hypothesis to be tested dealt specifically with the mode of presentation. It was stated as follows:

Ho4 - Students who are administered unit test items in sequential order by objective and no feedback will score significantly higher on final exams than students who are administered similar tests in random order by objective.

The data for the study again supported the hypothesis. Although the differences were much less than those related to feedback, they were still significant at the .05 level. It would appear that feedback is more important than modes of presentation to final exam scores.

Conclusions, Implications and Recommendations

The following conclusions, implications and recommendations are presented based on the findings of this study and from a systhesis of related studies identified as supporting the problem of test item feedback.

(1) Corrective feedback does slightly improve the achievement scores of students as measured by objective tests administered by computer.

The findings of this study imply that the more feedback provided to students in testing, that is providing the rationale for why answers are wrong or right, the better the learning and the longer it will be remembered. Retention of course materials is enhanced with the use of corrective feedback on test items. This would imply that students who receive corrective feedback on sample or instructional tests learn the material better and remember it longer. When teachers develop instructional materials, they should attempt to use this type of presentation with instructional materials or unit tests. On the other hand, test feedback may not be appropriate for initial competency testing of students. If teachers use competency testing to establish instructional procedures, partial feedback of right or wrong may be appropriate to establish a learning set for students.

(2) Computer generated exams appear to be most beneficial to students when they have control over the speed of item presentation and the objectives over which they are to be tested.

Informal discussions with students in this study and an analysis of previous studies which deal with attitudes of students toward testing show that the attitudes of students toward computer administered exams are better when students feel they have control over when tests are given and have some input over what objectives are to be examined. CMI system developers should consider providing these options to students when developing software authoring packages. The general attitude of students toward testing is improved when students feel some control over the timing, mode and management features of the testing structure. Teachers should establish what to teach and what to test and students should have input to when testing is done.

(3) Immediate feedback on specific test items is more beneficial to students than delayed feedback. Retention of materials is enhanced with immediate feedback.

Instructors who administer tests on the computer should provide immediate feedback to students. Any CAL system generating test items should have the provision for at least correct/incorrect feedback on tests as the test is administered.

(4) Low ability students seem to benefit more from corrective test feedback than do high ability students.

This research study looked at ability groupings of students and final test scores for the various modes and test feedback presentations. No significant difference was found by ability grouping when identified by ACT scores and final exam scores. Previous research studies, however, indicate that low ability students appear to gain more from test feedback than high ability students. This would imply that low ability students gain more benefit from corrective feedback than high ability students. Although the differences were not apparent in this study, they have been identified as significant in other research. More study needs to be done regarding the ability level of students and benefits of test feedback and modes of presentation as it relates to test administration and delivery by computer. The findings of this research, which states that all students do better on a test when item feedback is provided on practice unit tests, implies that teachers and instructional developers should implement a progress record as well as a feedback system within any computer administered test system. Knowledge of results is important to students as learning takes place but whether computer administered tests are equally beneficial to all students or more important to low ability students has yet to be specifically determined.

(5) Tests generated by sequential objective in random order by item yield higher scores than tests created by random objective order.

This conclusion implies that teachers should present information to students in a planned format generally by objective. If testing is to be effective and efficient, the presentation of questions by objective in sequential order will produce higher scores. If students are presented items on a test over a specific objective, they have a tendency to relate the information better than when test items are presented in random objective order. Developers of CMI test generation systems should take into consideration the ability of asking the system to generate tests in sequential/random order by objective and in random/sequential order by test item. With this configuration, the teacher will have the option of using the test presentation mode in either format. The general conclusion of this study is that students perform better on final tests if they have instructional tests with corrective feedback presented in sequential order preceding the formal final exam. Tests with items clustered by objective result in higher final exam scores than tests presented in random objective and random item order.

Teachers who construct sample or practice tests for students on the computer should use the feedback for incorrect answers as a means of supporting instruction. When students are learning information for course work, corrective feedback and problem explanations for incorrect answers are effective ways to extend an understanding of concepts.

BIBLIOGRAPHY

Arkin, Robert M. and Schumann, David W., "Effect of Corrective Testing: An Extension", (University of Missouri-Columbia). Journal of Educational Psychology, October 1984, Volume 76 (5), pp. 835-843.

Arkin, Robert M. and Walts, Elizabeth A., "Performance Implications of Corrective Testing", (University of Missouri-Columbia). Journal of Educational Psychology, August 1983, Volume 75(4), pp. 561-571.

Bender, Timothy A., "Improving the Use of Classroom Feedback", Presentation American Educational Research Association, 68th Conference, New Orleans, LA, April 23-27, 1984. Conference Proceedings.

Gallagher, Paul D., "An Investigation of Instructional Treatments and Learner Characteristics in Computer-Managed Instruction Course", Florida State University, CAI Center, Tallahassee, FL. Technical Report No. 12, July 30, 1970, 94 pp.

Hagerty, Nancy K., "Development and Implementation of a Computer-Managed Instruction System in Graduate Training", Florida State University, CAI Center, Tallahassee, FL. Technical Report No. 11, June 30, 1970, 161 pp.

Hail, Himmy.G., "The Effects Of Using Immediate Feedback Answer-Until-Correct Test Procedures On Retention", (Baylor University - Waco, Texas). Dissertation Abstracts International, January 1983, Volume 43 (7-A), #2285.

Hannafin, Michael J., "The Effects of Systemized Feedback on Learning in Natural Classroom Settings", (University of Colorado-Boulder). Educational Research Quarterly, Fall 1982, Volume 7(3), pp. 22-29.

Homa, Donald and Cultice, Joan, "Role of Feedback, Category, Size, and Stimulus Distortion on the Acquisition and Utilization of Ill Defined Categories", (Arizona State University-Tempe). Journal of Experimental Psychology: Learning, Memory, & Cognition, January 1984, Volume 10(1), pp. 83-94.

Jelden, David L., "Effectiveness of Computer Based Education: The UNC PHOENIX System". (University of Northern Colorado-Greeley, CO). Staff Study-1986 Available From ERIC (EDRS), 118 pgs.

Kulhavy, Raymond W., "Feedback Complexity and Corrective Efficiency". Contemporary Educational Psychology, Vol. 10(3), July 1983, pp. 285-291.

Lawler, R. Michael, "An Investigation Of Selected Instructional Strategies In An Undergraduate Computer-Managed Instruction Course". Florida State University, CAI Center, Tallahassee, FL. Technical Report No. 19, April 30, 1971. 107 pp.

Peeck, J. et.al., "Effects Of Informative Feedback In Relation To Retention Of Initial Responses", Contemporary Educational Psychology, October 1985, Volume 10 (4), pp. 303-313.

Pifer, Raymond E., "Effects Of The Use Of Feedback On Achievement", (University of Chicago). Dissertation Abstracts International, September 1981, Volume 42(3A), #916.

Plake, Barbara S.; Melican, Gerald J.; Carter, Linda; Shaughnessy, Michael J., "Differential Performance of Males and Females On Easy To Hard Item Arrangements: Influence Of Feedback At The Item Level", (University of Nebraska-Lincoln). Educational & Psychological Measurement, Winter 1983, Volume 43(4), pp. 1067-1075.

Schimmel, Barry J., "A Meta-Analysis Of Feedback To Learners In Computerized And Programmed Instruction", Presentation-American Educational Research Association, Montreal, Canada-April 11-14, 1983, 17 pgs.

Schroth, Marvin L., "Directive Concept Formation: S-R Hypothesis Versus Cognitive Theory", (University of Santa Clara-California). Journal Of General Psychology, January 1984, Volume 110(1), pp. 87-92.

Vitulli, William F., "Effects Of Immediate Feedback On Computer-Assisted Testing of ESP Performance", (University of South Alabama-Mobile). Psychological Reports, October 1982, Volume 51(2), pp. 403-408.

D. L. Jelden
Educational Technology Center
Univ. of Northern Colorado
Greeley, CO 80639
(303) 353-3892

The Computer Environment and the Teaching of Writing

Pam McRae and Mark Sherman

Western Washington University

Abstract

The use of computers in the classroom setting allows teachers of writing to circumvent a number of problems associated with more traditional methods of teaching their subject. This paper discusses some of the difficulties of teaching writing by the traditional method and explores the ways in which computers create new options for instructors by enabling them to intervene in a student's writing while it is in-process. The advantages of using software specifically designed for use in an academic setting and the overall impact of the computer environment on the teaching of writing is assessed.

The Computer Environment at Western Washington University

In the Fall of 1986, a grant from the administration at Western Washington University enabled the University's English department to set up an environment for the computer-assisted teaching of Freshman writing. Two rooms were equipped for the computer-assisted teaching of English 101, a general requirement for all university students. One room was equipped with 30 IBM PCs, printers, desks and tables. In another room a single IBM PC was hooked to a ceiling mounted Sony Video projector capable of displaying the output of the computer on a large screen. A number of word processing packages, such as Word Perfect and PC-Write, were reviewed by the English department for possible use in these classrooms. The selection committee decided on College Writer, a word processor developed by a software team at the University. Because the developers of College Writer had sought out and implemented the suggestions of English faculty regarding the needs of their students, this word processor is, in a sense, custom-tailored for use in the writing classroom. Because it requires an absolute minimum of computer knowledge to be operated effectively, English instructors are able to devote classroom time to teaching writing techniques rather than the mechanics of word processing.

After a full year of instruction in these rooms, it is clear that the computer classroom has had profound effects on the way in which writing is taught. In this paper we are interested in exploring some of the effects computers have had on the teaching of writing--in some cases enabling instructors to teach in ways previously unavailable to them--and in examining the way in which computers by their very nature initiate far-reaching changes in the way students learn and perceive the activity of writing.

The Traditional Teaching of Writing

Traditionally, the teaching of writing in the college setting has been managed as follows: students are lectured on the formulation of a reasonable thesis, the proper development of unified paragraphs, and the creation of topic sentences. Students are schooled in processes of revision and reminded of usage problems. Often, readings are assigned from anthologies containing examples of well-written essays, and students are asked to discuss these examples in class. After this, students are supplied with topics upon which to write and instructed to begin the drafting process at home. Often teachers make comments on the drafts of their students and offer suggestions on the areas in which a student can hope to strengthen his or her paper. Finally, the student is sent off to produce a final draft that will be handed in for a grade.

Some Difficulties of Teaching Writing with the Traditional Method

The traditional method's dependence on lecturing creates a number of problems for the instructor of writing. Writing--like an athletic sport--is a skill, the learning of a complicated process, and as such is not easily taught by exposition, by the lecture method. One can hardly imagine the coach of a basketball team expecting players to improve their performance on the court by listening to a lecture on the intricacies of the jump shot. To be successful, the coach of a basketball team must intervene at the moment of performance, instructing the player, for example, to lift his legs higher or to arch his back when making a certain shot. Recent literature on education has encouraged teachers of writing to employ the workshop-style method in their instruction and to see their function in the classroom as one of being a guide to students as they are engaged in the act of writing (Murray, 1985). However, until the introduction of computers and word processors into the educational environment, there has been no simple or entirely satisfactory way for teachers instructing large classes to do this, to intervene in an individual student's writing and demonstrate strategies while a piece of writing is in-process.

Writing for an Audience

One of the greatest challenges to the teaching of writing in the traditional classroom is convincing students that what they write in private must be designed for public consumption. Because in the traditional classroom the bulk of a student's writing is done at home and seen only by the professor, a student naturally tends to think of her or his writing as a private rather than a public act, a communication between two parties: the student and the teacher. Computers help the instructor address this problem.

A computer monitor is by its very nature more public than a piece of paper. Students, by merely turning their heads, can take in the writing on their neighbor's monitor at a glance. This feature allows an instructor to give an exercise--such as the creation of an introductory paragraph containing a thesis--and oversee a student's performance by strolling through the classroom. The effect of this is two-fold: it allows the instructor to make suggestions to a student while their writing is in-process and it tends to reinforce the concept that writing is a public act of communication. This method of instruction is, of course, not new. It is a method that has been used successfully, for example, by teachers of art in the studio-style classroom for years. However, until the introduction of computers into the classroom, this method has been difficult to emulate in the teaching of writing.

The Problem of Demonstration

Demonstration has always been a difficulty for the teacher of writing. A single paragraph in a five page essay can often run to fifteen lines of text. This fact makes it very difficult for the teacher of writing to display it effectively. Writing even a single paragraph on a chalkboard is not only time consuming but results are often vitiated by the legibility of an instructor's handwriting and by the problem of making revisions with chalk and an eraser. Because of this, it is often very difficult for the writing instructor to effectively demonstrate the idea of revision to beginning writers. Using a computer in conjunction with a video projector allows the teacher to legibly display several paragraphs of an exercise or an essay-in-progress. Changes can be made to this text--sentences can be removed or altered, entire paragraphs can be transported from one place in the essay to another--and instantly displayed. This ability to demonstrate is important because it can be used to convince students that an essay is made up of many smaller units--sentences, paragraphs-- that can be moved or altered in the service of communicating ideas powerfully and coherently. Since such revision is at the heart of effective writing, the ability to allow students to glimpse this activity in-process is very important. The computer classroom allows this to be done simply and effectively.

The Computer Environment Changes the Student's Concept of the "Paper"

In the past, an essay has been something that the student interacted with physically, with paper and pen, perhaps, or a typewriter. The name commonly given to student essays reflects this: the "paper." Once a student placed a sentence upon a sheet of paper it remained fixed there. Any changes to an essay meant that a new paper would have to be created. A student's paper could be held physically--clutched to one's bosom, so to speak--and for this reason tended to become in the student's mind something private and very personal--a possession. A student in the act of handing his or her paper to a fellow student for peer editing or evaluation was placing himself at risk. The paper was a personal possession on "loan" to a fellow student. The computer environment changes this significantly. In some cases, the student's essay will become a piece of paper only when it is finally printed before being turned in for a grade. An essay now exists on disk or on the computer monitor displaying the paper. This is, of course, far less intimate, far less of a possession than a piece of paper. In this case, the message of the computer environment to the student is that there is nothing final about the essay. The words and sentences on the screen are in a fluid state, susceptible to being rearranged, deleted, or added to at the student's whim. The fact that a paper is stored on a disk, and for that reason cannot actually be seen, transforms it into a possibility, a potential. The computer environment, thus, echoes points that the teacher of writing attempts to inculcate in students: writing is a process that begins as an idea in the student's head and goes through countless revisions, drafts; and does not become a "product" until its submission to the instructor for a grade.

Computers--a Sterile Environment?

Inevitably, the fact of so many computers in a single classroom causes the following question to arise: will students (and even teachers) be intimidated by the presence of so much high-technology? In the case of Western's computer-assisted writing program, we have found this not to be the case. College Writer, the word processing software used in these writing courses, is custom-designed for the student writer in the university classroom. It is rapidly learned and quickly becomes an invisible part of the learning environment. The effect of this on first-time users is striking. In many cases, by the end of the second classroom session students are creating papers using the computer. Because computer literacy possesses such a high value in our culture, the fact that a student can learn to operate this word processor quickly works to affirm a student's self-esteem.

The Message of the Environment

The presence of computers in the classroom not only gives the writing teacher options previously unavailable, but also affects the ways in which his or her students tend to view the act of writing. Here at Western, we have found that the use of computers produces more interaction in the classroom, makes students feel better about the act of writing, and, thus, causes them to write more. The computer environment makes emphatic a message that has been emphasized continually by writing teachers: writing is a process; we learn by doing. Since, in this case, what is being taught--writing--is a skill, this emphasis is an appropriate one.

References

Murray, D. M. (1985). A Writer Teaches Writing. Boston: Houghton Mufflin.

Pamela McRae
Tutorial Center
Old Main 380
Western Washington University
Bellingham, WA 98225
(206) 676-3855

Technology and Group Instruction:

A Communication and Management Tool for Teachers.

Steven L. Robinson, Ph.D.
Educational Technology Research Program
Amherst H. Wilder Foundation

ABSTRACT

One major problem with current forms of technology-based instruction is that the role of the teacher as a manager, presenter, and monitor is limited. Another problem is that applications for group instruction have been all but ignored. This article will introduce a technology-based, communication and information management system designed for the teacher operating in the context of group instruction.

Teacher coordinated group instruction is an inescapable reality of educational enterprise. As one noted educator commented, in almost every school, the total group is the vehicle for teaching and the teacher is the pivotal figure in the group.

Classrooms, however, are complex, multidimensional, layered environments. Teachers must organize 20 - 30 students into working units that change over time. They must select and arrange activities, present material, provide opportunities to respond, monitor and pace instruction in a manner that accommodates each learner's needs. In the best of circumstances, this is a difficult task essentially because groups create distance between the teacher and individual students. This distance prevents teachers from ensuring adequate student engagement, student-teacher interaction, monitoring of work and feedback.

Instructional methods such as one-to-one tutoring and computer-assisted instruction attempt to overcome the problems imposed by groups by increasing learner access to teaching resources outside the group context. As effective as these methods may be, they cannot for practical and cultural reasons, replace group instruction. Due to limited resources, neither is practical on a large scale and the social nature of human enterprise makes teacher-directed group instruction desirable even in the face of potentially more effective methods. As Calfee (1985) said:

> Teachers play a vital role in the educational process, in large measure because an important part of becoming educated has to do with interpersonal relations- with learning to communicate and join in problem solving with others (p. 12).

Therefore, groups are an important and dominant context for instruction that cannot be ignored. Although methods for improving the efficiency and effectiveness of teachers in classrooms do exist most do not involve technology. Perhaps technology may be brought to bear on these issues. The qualities of computers as information processors might be used to enhance the process of interaction between the teacher and his/her students. In this manner technology might better serve education's goal of appropriate instruction for all students.

A Technology-based, Group Instruction Alternative

The Discourse Educational Communication System (or "Discourse" for short) represents an application of technology to the limitations of group instruction. Discourse combines the power of the computer as an information processor which can calculate and store, with the power of a teacher as a knowledge processor, who can reason and inform (Feig-

90

enbaum & McCorduck, 1983). The system facilitates and processes student responses and makes this information available to the teacher. This provides the substance for teacher diagnosis, prescription, presentation, monitoring and feedback; the critical teaching functions which influence student learning (Fisher, Berliner, Filby, Marliave, Cahen & Dishaw, 1980). In this manner Discourse keeps the teacher at the center of instruction and employs a computer to overcome the limitations to effective teaching imposed by a classroom of 20-30 students.

Discourse keeps the teacher at the center of instruction by electronically channeling student academic responses to a computer monitor viewed by the teacher. The system fundamentally operates as a communication tool. Each student in a class is supplied with a response device consisting of a standard "QWERTY"-type keyboard and an L.E.D. display line. These "Studycoms" are connected to one computer with a color monitor, printer and keyboard used by the teacher. Operating software permits the teacher to control and coordinate instruction. First the teacher prompts students to "sign-on". As they type their names on the Studycoms, the names appear on the teacher's monitor in a column by the students' seat locations. The teacher may then prompt students to respond to spontaneous or prepared questions whether presented orally, by audio-visual media, or by workbooks or worksheets. Students type their answers on the Studycoms. These answers are simultaneously displayed on the teacher's monitor by the student's name as shown in Figure 1. With more than 20 students, the teacher may page to another screen of names. The teacher uses the information displayed to provide corrective feedback, explain, reteach, or whatever is appropriate for the situation.

Discourse helps teachers accommodate instruction to individual needs in the classroom environment in several ways. The system 1) provides facilities to prepare and conduct teacher-paced or student-paced lessons, 2) increases students' opportunities to respond under the teacher's direct supervision, 3) increases opportunities for immediate corrective feedback, 4) provides comprehensive, efficient and timely record keeping facilities, and 5) permits the control and integration of a variety of audio-visual media for presentation purposes. Each of these areas will be discussed separately.

Lesson Preparation

As briefly mentioned above, when Discourse is used, the teacher, not the computer, delivers instruction to the students. The system collects, organizes, processes and records students' responses. There is no pre-packaged instructional software. Content comes from regular curriculum materials or materials provided by the teacher. The teacher, by use of a simple authoring system incorporated as part of the operating software, prepares lessons for storage on a floppy disk. With this facility, teachers may prepare lessons which include 1) answers that the system will use to score student responses, 2) instructor notes (e.g. "Show the overhead on cell division"), 3) instructions for the automatic operation of audio-visual media or 4) the questions to be asked. These segments are associated with numbered frames (e.g. 01, 02, etc.). A set of frames constitute a lesson which is named and stored on a lesson disk for repeated use. This facility allows teachers to build their own lessons, then use, reuse, or revise them as appropriate.

Discourse, however, may be used without a lesson in computer memory. Used simply as a communication tool, teachers may prompt students to respond to questions provided by any means. The system displays the students' answers on the teacher monitor.

Discourse lessons may be paced by the teacher or by the student. For teacher-paced lessons, the teacher controls the sequence and timing of student responses. For student-paced lessons, students are provided with numbered worksheets. Answers to the items are entered on the Studycoms; the students control the sequence of items attempted and the time spent on each item.

The system also provides the capability of organizing multiple group student-paced instruction. Discourse will process the work of up to seven groups working on seven different lessons simultaneously.

Opportunity to respond and teacher monitoring

Opportunity to respond under a teacher's supervision is a critical component of achievement (Delaquadri, Greenwood, Whorton, Carta and Hall, 1986). The Discourse System provides a response device for every student. When the teacher asks a question, all students have an equal opportunity to respond. These responses are organized and displayed at the teacher's monitor for efficient processing. This creates an environment with high levels of guided practice and enhanced accountability. Guided practice and accountability promote student engagement, another consistent correlate with achievement (Stallings and Krasavage, 1986).

Students responding via Discourse are not restricted to single letter or number responses. Students may construct responses with up to 186 characters and may therefore answer "how" and "why" open-ended type questions (Figure 1). They may also edit their responses with a backspace and linefeed Studycom facility. Whatever the nature or length of the response, and regardless of whether the lesson is teacher-paced or student-paced, the response is displayed for viewing by the teacher. This permits the teacher to immediately direct feedback and correction to individual students most in need of help.

Corrective Feedback

Immediate and direct verbal feedback from the teacher is only one form of feedback facilitated by the Discourse System. For lessons in computer memory that contain answers (such as tests or worksheets), the system will scan each student's response and automatically provide instant response contingent feedback to individual students in the form of two flashing Studycom lights . A beep tone may also be optionally paired with the lights. When the lesson is created, the teacher specifies which answers will be paired with feedback. These selections may be overridden at any time during the lesson.

Another form of feedback capability controlled by the teacher is the "public" display. The teacher may, even while students are typing, type a student's seat location to display his/her individual response. This response appears for the entire class on a large video monitor (not the teacher's monitor) that is also used to display other media . The teacher may display any number of responses, one at a time.

Record keeping

The Discourse System makes a wide range of recordkeeping facilities available to the teacher. These provide an additional form of performance monitoring which aides the diagnostic and prescriptive functions of teaching.

Three report types (detail, question and student) are instantly available during, at the conclusion, or after a lesson or test. The teacher may access these reports without disturbing students as they respond. Detail reports list the performance of each student on each question, scoring the items as correct, incorrect, or not attempted. Question reports list, in percentage form, the performance of the class as a whole on each item. Student reports describe by percentage the performance of individuals on the lesson as a whole. Each report may be dynamically displayed on the teacher monitor or may be printed. One option for detail and student reports prints individual reports for each student. The teacher may therefore administer a test and distribute individual, hard copy results before the end of the period. This saves valuable time normally spent correcting papers and gives students more timely information about their performance.

The teacher using Discourse may elect to record student responses and scoring data on a floppy disk. At the end of a day, during a preparation period, or even with a computer at home, this information may be recalled. The teacher may reevaluate responses, enter critical comments and generate individual reports which include these comments. Other software automatically accumulates scores on lessons, tests or individual objectives so that student performance may be tracked over time.

Audio-visual Media

Audio-visual media has excellent potential to complement group instruction but is underutilized in classrooms. Media devices typically must be obtained from a media center and require some technical knowledge for the operation of each device.

Discourse helps overcome these limitations by permitting the teacher to control a random access slide projector, a video cassette recorder and a videodisc player from the computer keyboard. Any off-shelf slides, video cassettes or videodiscs may be used.

"Digital" slides generated with a graphics tablet and stored on floppy disks by the teacher may also be accessed by the system.

The control of each media device is accomplished by pressing a function key (e.g. "F7" for the VCR). This produces a menu explaining the control options such as "Press enter for play" or "Press F10 for stop".

These devices may also be automatically controlled by instructions within disk-stored lessons. For example, the item "01;P25PV15000,16500V" will display slide #25 and will play from frame 15000 to frame 16500 on the videodisc. This capability creates the potential for high quality, interactive, media-based presentations designed for groups which may be used repeatedly. The teacher is freed from concerns about how to present material and may consequently devote energy to teacher-student interaction and the adjustment of instruction to meet individual needs.

Research with Discourse

For the past two years, with financial support from the 3M Foundation and the cooperation of Dr. Jack Zawels, the system's inventor, and the St. Paul School District, the Amherst H. Wilder Foundation has investigated the instructional effects of the Discourse System in classroom settings.

This research indicated the system may be a potent instructional tool. In one study, after 45 minutes of training on how to use the system, a teacher was able to produce over a 100 percent increase in the percentage of academic responses students made during regular instruction (Robinson & DePascale, 1985a). In another study with severely learning disabled elementary school students working on long division worksheets, accuracy increased from 70 to above 90 percent when the immediate feedback facility (flashing lights and beeps contingent on a correct response) was incorporated as part of the lesson (Robinson & DePascale, 1985b). This and other research indicates the Discourse System may be a useful tool for teachers to accommodate instruction to the needs of individual students in group settings. The critical mechanism appears to be that all students within a class have more opportunities to make academic responses under a teacher's direct supervision. A program of expanded research over the next three years will continue to explore the system's effects.

Summary

In summary, this article describes an innovative application of technology to the problems of group instruction, the dominant environment for educational enterprise. This technology, called the Discourse Educational Communication System, places the teacher at the center of teaching in a manner that enhances control over important components of effective instruction such as opportunities to respond, student-teacher interaction, performance monitoring and feedback/correction. The focus of this technology on these components results in the implication that the system may be beneficially applied to any training or educational setting where the group is the context for instruction.

References

Calfee, R. (1985). Computer literacy and book literacy: parallels and contrasts. Educational Researcher. 14(5),8-13.

Delaquadri, J., Greenwood, C. R., Whorton, D., Carta, J. J., Hall, R. V. (1986). Classwide Peer Tutoring. Exceptional Children. 52(6),535-542.

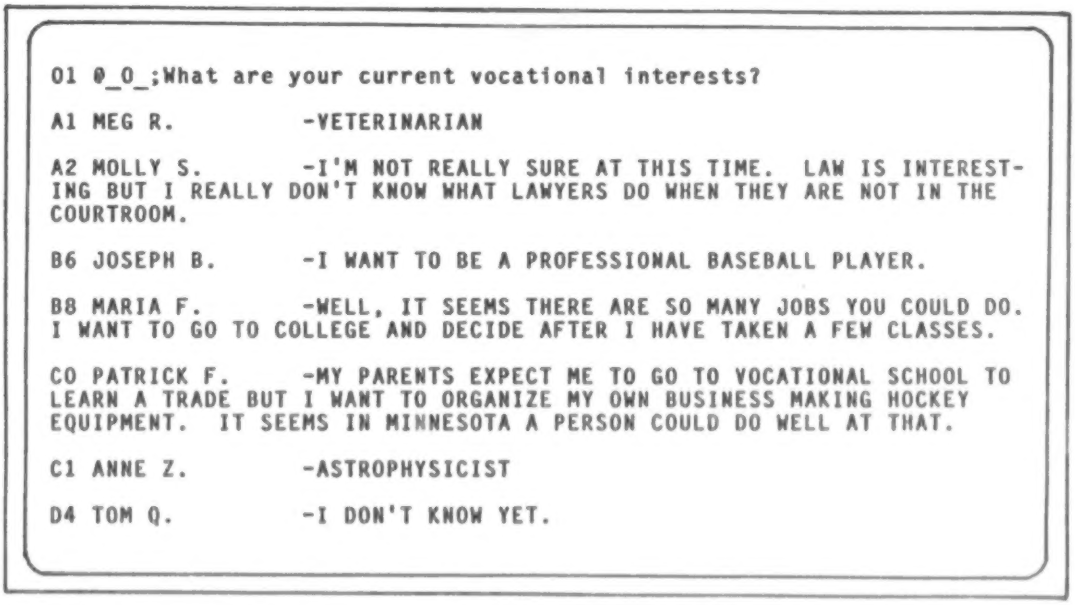
Feigenbaum, E. A., McCorduck, P. (1983). The Fifth Generation. Reading, MA: Addison-Wesley.

Fisher, C. W., Berliner, D.C., Filby, N.N., Marliave, R., Cahen, L.S., Dishaw, M. M. (1980). Teaching behaviors, academic learning time, and student achievement: An overview. In C. Denham and A. Leiberman (Eds.), Time To Learn. Washington, D.C.: National Institute of Education.

Robinson, S. L., DePascale, C. (1985a). Effects of a technology-based communication system on student responding and teacher questioning. (Research Report No. 9) St. Paul, MN: Amherst H. Wilder Foundation.

Robinson, S. L., DePascale, C. (1985b). The effects of computer-delivered feedback on long division performance of learning disabled students. (Unpublished research report). St. Paul, MN: Amherst H. Wilder Foundation.

Stallings, J., Krasavage, E. M. (1986). Program implementation and student achievement in a four-year Madeline Hunter Follow-Through project. The Elementary School Journal. 7(2),117-138.



```
01 @_0_;What are your current vocational interests?

A1 MEG R.          -VETERINARIAN

A2 MOLLY S.        -I'M NOT REALLY SURE AT THIS TIME.  LAW IS INTEREST-
ING BUT I REALLY DON'T KNOW WHAT LAWYERS DO WHEN THEY ARE NOT IN THE
COURTROOM.

B6 JOSEPH B.       -I WANT TO BE A PROFESSIONAL BASEBALL PLAYER.

B8 MARIA F.        -WELL, IT SEEMS THERE ARE SO MANY JOBS YOU COULD DO.
I WANT TO GO TO COLLEGE AND DECIDE AFTER I HAVE TAKEN A FEW CLASSES.

C0 PATRICK F.      -MY PARENTS EXPECT ME TO GO TO VOCATIONAL SCHOOL TO
LEARN A TRADE BUT I WANT TO ORGANIZE MY OWN BUSINESS MAKING HOCKEY
EQUIPMENT.  IT SEEMS IN MINNESOTA A PERSON COULD DO WELL AT THAT.

C1 ANNE Z.         -ASTROPHYSICIST

D4 TOM Q.          -I DON'T KNOW YET.
```

FIGURE 1. Discourse teacher monitor screen display showing item number, question to be asked, students' seat locations, names and responses to an open ended question.

Steven L. Robinson, Ph.D.
Educational Technology Research Program
2265 Como Ave, Suite 200
St. Paul, MN 55108
612-645-4679

YOUNG CHILDREN AND COMPUTERS: A LOGO PROJECT REVISITED

DR. CHARLOTTE SCHERER

BOWLING GREEN STATE UNIVERSITY

ABSTRACT

A Logo project, now implemented a third time, has been extensively revised in the light of new research on young children and computers. Descriptions and results of changes made are described and explained. Appropriate supplemental software and the role of university early childhood majors who are learning how to introduce computers to preschoolers is also discussed.

INTRODUCTION

Introducing young children to computers takes a great deal of forethought about the selection of programs and the appropriate format in which to include the computer activities. Ideally, according to the Highscope curriculum for successful preschool programs, computers should become just another of many possible activities from which children are able to choose (Keynotes, 1986). There are very good programs on the market designed for preschoolers and with careful selection and previewing a preschool staff can find appropriate programs for their children. However, as was found in some studies, all children do not like the same programs and all children are not ready to sustain interest for ten or fifteen minutes of time working with a computer program. Also, some children need much more direction from an adult than others. (Keynotes, 1986; Killian, Nelson & Byrd, 1986). Therefore, a variety of programs will need to be available and a sufficient adult/child ratio must exist.

THE BGSU LOGO PROJECT

When two professors in the College of Education and Allied Professions at Bowling Green State University wanted to train early childhood majors by providing some "hands on" computer experience with preschoolers, a project was designed using Logo as the primary thrust and supplementing with several other programs that emphasized alphabetic, numeric and problem-solving skills. It was believed that Logo would involve thinking skill and decision making processes, while the other programs would reinforce concepts and skills that were already being taught at the preschool that would be a necessary prerequisite for beginning reading and math. Since we were bringing the children to the computer lab at the university, it was not entirely feasible to make the computers one of many activities, but a good balance between on- and off-computer activities was provided (Scherer, Spring 1986). When the project was started, little was in the literature about working with young children and computers, except for the idea of using a single keystroke version of Logo. However, as the project was being implemented, more appeared in the literature (Keynotes, 1986: Goodwin, Goodwin, Nansel & Helm, 1986: Killian, Nelson & Byrd, 1986), and this author (the Project Director) also heard papers at national conferences regarding the use of computers with preschoolers. Hence, following a successful pilot project and then a two semester sequence project with early childhood majors introducing the computer to preschoolers through the use of Logo and other appropriate programs, the project was extensively revised and offered again as a more organized and sophisticated two semester sequence of learning for early childhood majors and preschoolers.

In the earlier version, the children came for a 45 minute period, divided roughly into three sections: a fairly formal instructional period, a physical activities period and a "hands on" period. Logo was introduced using F, L, R, D, and U from the Terrapin "instant" Logo. (Terrapin, 1982). One at a time, other programs were introduced and the children used them after they had done the Logo project for the day. It was only after several sessions that the children had much choice of program on a given day. This was effective, but did not fit readily the idea that some children will favor one kind of program more than another, and that children need to be making their own choices. Also, for the most part we had one-on-one adult/child ratios. Using an observation scheme (Clements & Nastasi, 1985), we discovered that instead of becoming more self-reliant and independent, the children actually became more dependent over a period of five weeks. The relationship between early childhood major and child was affectionate and very close. It was felt that this was good neither for the child (who tended to become less self-reliant), nor the university student (who needed to realize that such a close relationship was neither productive nor realistic). So, some revisions were made when the project was set up for fall 1986.

REVISIONS IN THE PROJECT

A new version of single keystroke Logo was used (Birch, 1984). This time in addition to F, L, R, and U, B for back, C for curve (a half circle) and S for square were included. Instead of D which made little or no sense to the children, E was used for erase. The addition of C and S made it possible to make more creative designs than was possible earlier. Our favorite story is the one of the little girl who was drawing away using C fairly frequently. Eventually, she used two C's in a row near the top of the screen and of course one went off the screen and appeared at the bottom. She exclaimed, "Oh, this was going to be a snowman, but it broke!" Another change was the addition of color. We used four numerals 2, 3, 4, and 5, for colors and then with the availability of Printshop which supports color, we were able to use Screen Magic and print out their creations in color. This delighted the children.

Instead of introducing the other supplementary programs slowly, one at time, we now introduced a number of them very early in the set of sessions. Although we continued to emphasize Logo on a regular basis and encouraged at least one Logo picture before they went on to other things, they now had free choice of five different programs during each "hands on" session. The other programs were: MECC PreReading/Counting, Muppet Keys using Muppets on Stage, MECC Right of Way, and Moptown Parade. They liked all of them in general, but they often had to take turns with Muppet Keys, since there were only two of these and they were very popular. We chose MECC Right of Way because it deals with concepts of left and right and we thought it might help with this difficult concept. It was a semi success. The problem we quickly discovered was that this program has children using any key from the right side of the keyboard to indicate an object on the "right" and any key from the left side of the keyboard to indicate "left". Since we had just spent a fair amount of time helping them identify the "R" key (on the left side of the keyboard) for a "right" turn and the "L" key (on the right side of the keyboard) for a "left" turn of the turtle, it was confusing to then work with a program that did the opposite. However, the monkey on Right of Way is appealing as is moving cars into parking spaces and the children by and large were able to overcome their confusion. Many often chose this program as one of their activities. We have found only the first three games on the Moptown Parade suitable for the four year olds and most of the MECC PreReading/Counting games require a fair amount of adult supervision, at least at first. Muppet Keys have the distinct advantage of being almost totally usable by the child independently.

To encourage more independence and self-reliance, in this version of the project, we assigned different university students on different days to oversee two and sometimes three children. We rotated the "extra" university students into completing an observation scheme. The purpose of this was twofold: 1) it was felt that the students would gain from some systematic observation of the children where they could focus on the behavior of specific children, and 2) it was a way to collect informal data which we could analyze regarding the children's behavior. One positive result of this approach was that the children, not having a one-on-one situation with an adult, were much more eager and ready to go and see what another child was doing or to work with another child playing the game in a cooperative manner. Another positive result was that the children attempted to solve some of their own problems, rather than have the adult jump in and help before trying for themselves. Although there were days and particular programs that demanded a smaller adult/child ratio, we were able to be flexible on this matter and overall, it did work better to have the ratio of children to adults increased most of the time, especially after the first two or three sessions. However, assigning one student to oversee three children was not very effective.

Another change was the culminating Logo project. Because the project was scheduled for later in the semester, the jack o'lantern used the first time around was no longer appropriate. So, a snowpeople project was devised. This had the advantage of teaching the university students an important point in sexism and avoiding religious issues with Christmas activities. The idea was that two snowpeople, a snowlady and a snowman were drawn on the screen. The children were to use single keystroke commands to move the turtle to a place where they wanted to place a button. They could also choose a color for the button. Some had snowpeople covered with buttons, others only placed one or two, but all were successful with this project with appropriate supervision from the adults.

We also made a change concerning the training of the university students. They were taught Logo and all the other supplementary programs. In the process, they reviewed one of the supplementary programs by completing an evaluation form. One question on the form asked how suitable they thought it might be for the children. They completed this same questionnaire after the completion of the project, using the same program as before, but this time completing it in light of how they saw the children reacting to the program when it was actually used in the project. The results of this were interesting. There were not many differences in their pre- and post-assessments of the programs. Only in a couple of cases did they decide that the program could not be used as independently by the child as they first thought. The five programs that ended up being evaluated by this group were Stickybear A,B,C's, Moptown Parade, MECC Pre-Reading/Counting, Muppet Learning Keys and Teddy's Playground. All indicated the child enjoyed the programs, but one stated that Stickybear A,B,C's did not sustain interest for very long. The students felt that Moptown Parade, MECC Pre-Reading/Counting and Teddy's Playground were the least usable independently by our four, nearly five year olds. By having the students evaluate the programs before and again after they had worked with the children, the students were helped to focus on the skills to be learned from each program and it made them aware of the importance of seeing how the children reacted to each program.

In the fall, we also asked the students to plan for leading most of the group physical activities as preparation for taking on greater teaching responsibilities of leading group off-computer lessons in the spring. This was planned as a direct result of having given them too little responsibility in the fall and too much in the spring the first time around.

When the children return this spring, we will introduce several other programs, one that is suitable for use with the Muppet Keyboard. This will be the Muppet Word Book by Sunburst. Other programs will include Sunburst's Space Waste Race, more use of Teddy's Playground, Early Discoveries, a program dealing with patterns and sequences, Getting Ready to Read and Add and Stickybear Town Builder. We will also move them into some regular Logo, and try to teach concepts of 90 and 45 degree angles. To provide practice with regular Logo, a game of helping the turtle find a lost shell (Teaching and Computers, 1986), will be used. In the spring the culminating project will once again be a flower which needs some dirt or a pot. They are to use regular logo commands to draw the dirt or pot. Obviously the pot is a real challenge for the most capable, but all can draw some dirt by making a horizontal line. This project proved successful last year and will be tried again.

Logo music has also become a standard part of the second set of sessions. The children can press letters (C,D,E,F,G,A,B,C1) to make their own music, and they learn to sing a song with the computer accompanying them (Miller, 1986). In the fall, they learned several songs about the turtle with words set to familiar tunes such as "I've Been Working with the Turtle" (I've Been Working on the Railroad) and "I'm a Little Turtle, Small not Frail" (I'm a Little Teapot") and "Mary Had a Turtle, Too" (Mary Had a Little Lamb), so the Logo music is a logical extension.

Stories and poems are also part of the sequence. The Little Turtle (who lived in a box, swam in a puddle, climbed on the rocks) was a popular fingerplay and the familiar Yertle the Turtle by

Dr. Suess and a not so familiar American folktale, Tortoise's Dream, held them spellbound for nearly fifteen minutes one day during the fall. The Tortoise and the Hare, another fable and The Monkey and the Turtle will be used in the Spring.

Physical activities include such things as jumping on and off specific letters on a large keyboard, playing The Turtle and His Friends to the tune of "The Farmer in the Dell" and Turtle Says (Simon Says) or pretending to move like a variety of different animals, including a turtle, of course, and will be used both semesters. Activities revolving around the snowpeople and flower projects are also used.

SUMMARY OF RESULTS

The questionnaire given to the students at the completion of the first semester of the revised project indicated the following:

a) Six of the eight who responded were "neutral", "disagreed" or "strongly disagreed" that they were comfortable using a computer themselves at the beginning of the project.
b) Likewise, eight of eight were "neutral", "disagreed" or "strongly disagreed" that they were very comfortable using a computer with preschoolers.
c) Seven "agreed" or "strongly agreed" that by the end of the project they were very comfortable using a computer with preschoolers. This indicated that two of our goals had been accomplished.
d) However, only half "agreed" or "strongly agreed" that it is very worthwhile to use computers with young children, so we did not succeed in demonstrating the value of computers for young children as we had planned.
e) But, seven of the eight who responded "agreed" or "strongly agreed" that computers can be used to reinforce concepts and skills that are taught by other methods, so they did see some value.
f) Only half "agreed" or "strongly agreed" that computers can help teach concepts and skills that can't easily be taught by other methods. We apparently did not succeed in demonstating the uniqueness of the computer to teach some skills.
g) Half "agreed" or "strongly agreed" that the assistance they received in training and individually was adequate to prepare them to work with preschoolers and computers. Our training time is limited and they need more practice themselves before working with the children, but it is difficult for them to see this until they are in action with the children. Perhaps they will be more secure and comfortable the second time around, when the children return Spring semester.
h) In terms of recognizing and being able to match descriptions of programs used with their titles, five of eight were able to correctly match all nine programs, two match seven correctly and one matched six. This was an improvement over last year's program and we were pleased that at least most recognized which skills were being developed by which program.

The students' written comments indicated that they felt that the balance between on and off computer activities was good, and they were positive about the learning taking place for both student and child. One even suggested that the university students should teach more of the group lessons. They will have an opportunity to do this the second semester, but we felt they needed the modeling experience first, as well as time to get used to working one on one before we gave them increased responsibilities of teaching more structured lessons to the whole group. During this second semester, the observation scheme, evaluation of new software and the questionnaire to university students will be used again to get results from this part of the project which can be compared with the first semester and with results from last year.

All in all, the changes we made for this year's project were well worthwhile. Results from the questionnaire to the students as well as our own observation indicate that our project has become increasingly organized and sophisticated as it has evolved over the past several semesters and we are accomplishing many of our goals of making the students comfortable using a computer and using a computer with young children, of making the students familiar with several programs suitable for young children, and of introducing the children to the computer. However there is still room for improvement and we still need to do a better job of helping the students learn to work with the children and the computers. Not all the students were convinced that computers are very worthwhile for young children, and not all felt that we adequately prepared them for their work with the children and the computers, but hopefully the steps we take during spring semester will change this.

REFERENCES

Birch, L. (1984). Introducing Logo to Primary Children (Computer Program and materials). Boston, MA: Houghton Mifflin.

________(January 1987). Computers as Assessment Tools - Using Computer Play and Exploration Activities for Child Assessment. Keynotes, 2(8), 1-6.

Goodwin, L., Goodwin, W., Nansel, A. & Helm, C. (Fall 1986). Cognitive and Affective Effects of Various Types of Microcomputer Use By Preschoolers. American Educational Research Journal, 23(3), 348-356.

Grimm, L. (1981, 1983). Moptown Parade (Computer Program). Menlo Park, CA: Addison-Wesley Publishing Company, Inc./The Learning Company.

Henson Associates, Inc. Trademark (1984). Muppet Learning Keys (Computer Device and Program). Pleasantville, NY: Sunburst Communications.

________(October 1986). How Young Children Use Computers: What Classroom Observations Tell Us. Keynotes, 2(5), 1-3.

Killian, J., Nelson, J., & Byrd, D. (August/September 1986). Child's Play: Computers in Early Childhood Programs. The Computing Teacher, 13-16.

Miller, D. (March 1986). Let's Make Music! Teaching and Computers, 3(6), 38-43.

________(January 1986). Mr. Turtle Goes to Kindergarten. Teaching and Computers, 3(4), 41-44.

Optimum Resources, Inc. Trademark (1984) Stickybear ABC's (Computer Program). Distributed by Weekly Reader Family Software.

Optimum Resources, Inc. Trademark (1984). Stickybear Numbers (Computer Program). Distributed by Weekly Reader Software.

Optimum Resources, Inc. Trademark (1984). Stickybear Shapes (Computer Program). Distributed by Weekly Reader Family Software

________(1981). Pre-Reading/Counting (Pre-2) Vol. 7, Version 1.0 (Computer Program). Minnesota: MECC.

________(1986). Right of Way (Computer Program). Minnesota: MECC.

Scherer, C. (1986). Early Childhood Majors Teach Logo to Preschoolers. In the Proceedings of LOGO '86, Cambridge, MA, 96-98.

Scherer, C.(Spring 1986). Pre-Service Teachers Use Logo with Preschoolers. MACUL Journal, 1-10.

Scherer, C.(February 1986). Pilot Project: LOGO for Preschoolers as an Aid for Teacher Preparation in Technology. In the Proceedings of the 27th International ADCIS Conference, New Orleans, LA, 39-42.

________(1982). The Terrapin Logo Language(Computer Program). Cambridge, MA: Terrapin, Inc.

________(Fall 1986). The Value of Child-Initiated Learning. High/Scope Resource, 3-6.

Vaidya, S. & McKeeby, J. (1985). Developing a Logo Environment in the Preschool. In Maddux, C.(ed)., Logo in the Schools, New York: Haworth Press, 81-89.

________(September 1986). What is "Developmental Education"? Why Is It So Important to Young Children? How Can Computer Learning Support It?. Keynotes, 2(4), 1-3.

________(Fall 1986). What Makes a Good Preschool Program?. High/Scope Resource, 7-9.

Dr. Charlotte Scherer, Director
Clinical and Computer Labs
215 Education Building
Bowling Green State University
Bowling Green, OH 43403
(419) 372-7392

Teaching Modular Hierarchical Programming Using LogoWriter™

Sylvia A. Shafto* and John D. Lawson

Towson State University, Baltimore, MD

Abstract

Students learning to program using LogoWriter can learn to write well-modularized, hierarchically organized programs by following a few model programs as examples. Preliminary research in a Computer Science and Society course suggests that after a few hours of laboratory experience, students were able to write relatively long programs that showed good use of procedures and control flow.

Students' (n=101) understanding of flow of control in Logo was evaluated using a pencil and paper Logo quiz, and students' prior experience with computers was determined a questionnaire. The Logo programs of a sub-sample (n=26) of students who successfully completed the assigned Logo programming project and who had no prior programming experience were evaluated for use of procedures and nature of control flow.

A large correlation matrix for the Logo quiz scores, responses on the questionnaire, and results of the program evaluation showed little predictive power for most of the variables.

Analysis of the use of procedures and nature of control flow suggests that students generalize from a few examples about using hierarchical control flow (procedures calling procedures), procedures of reasonable length, and iteration (REPEAT commands).

Teaching students about control flow in a course such as this is more important than teaching them about manipulation of data. Logo is a good language for this purpose since students learn to manipulate the data (turtle graphics) and write hierarchical procedures relatively easily.

Introduction

Breaking a large problem down into smaller ones is generally recognized as an essential component of effective problem solving, as well as an essential component of computer programming. Students who are learning to write a computer program for the first time are typically taught to write simple programs which do not include any procedures. Once the students are familiar with most of the language syntax necessary to produce any program, the use of functions and procedures is introduced. By that time it may be difficult to persuade the students to give up their "run-on" coding style. The virtues of writing complex programs in a modular and hierarchical style may not be obvious to the student. In contrast, students might learn to use procedures in the first assignment and study examples of programs which include only short sequences of commands. Given this model, will the student learn from the beginning to break the task into small units? It may be that students will learn to develop their programs using short procedures organized hierarchically--if they see only well written programs.

The design of the Logo language makes learning to write procedures very easy. The language uses many built in procedures such as "FD" to make the turtle move forward and "MAKE" to bind a value to a variable. Logo is also an extensible language. The programmer can create new procedures that are indistinguishable from Logo's original set of procedures. The syntactical use of the procedure name is precisely the same as the use of built-in procedures in the language. Typically teaching students to use procedures in Logo includes making it clear to the students that this similarity exists. The programmer is simply adding to the language's repertoire of commands when she writes a procedure.

---

* currently at San Jose State University, San Jose, CA

The research question asked in this study is: Can students who are without previous programming experience learn to write computer programs with procedures in a well modularized hierarchical design, if they use a language which has easily defined procedures, and they are introduced to well structured design early in their exposure to the language.

Research Design

Sample

The sample in this study included students taking Towson State University's 1986 fall semester course: Computers and Society. John Lawson and Sylvia Shafto taught the course. About two-thirds of the students were in lecture/laboratories where no more than 24 students were enrolled in a section. The remaining third were in a large lecture class (70 students) and had a graduate student teach them the laboratory material. The laboratory classes for the large lecture class were in smaller sections of 24 students.

The students enrolled in this class come from a wide variety of academic interests. The course is popular option for one of the "General University Requirements" in the sciences. The range of majors includes almost every area of study in the University. This group of students does not include many business majors since those students take a different, but similar course, oriented towards business applications of the computer. The students vary in their ages and status as students, including returning students as well as traditional resident undergraduate students.

Materials

LogoWriter software by Logo Computer Systems Inc. was used running on *Apple //e* computers by Apple Computer Inc.. The language uses a metaphor of the programmer as the writer of a scrapbook. The scrapbook is the *LogoWriter* diskette. The representation of file organization into a more recognizable context helps the student to concentrate on programming in Logo rather than first learning an operating system and the difference between DOS and Logo. The student and teacher can talk together about *LogoWriter* commands without the possible confusion from operating system commands. The scrapbook has a table-of-contents of pages. The pages are individual files--the table of contents is a directory listing of the files. Each page has a front side and a flip side.

The front side is an immediate mode programming environment. Two windows appear on the front side; a small command window where all typed immediate Logo commands appear and a larger output window where all output from executed turtle graphics commands and print commands appear. The cursor appears in the command window awaiting the input of commands from the programmer. Procedures cannot be written in the immediate mode, although they can be called and executed by typing their name in the command window. The cursor can be moved to the output window. Text typed in the output window is treated as a "label" for output in the window.

The flip side also has two windows; the small command window and a larger text window where procedures can be written. This text area behaves similar to a word processing program with word-wrap; cursor control with cursor arrow keys; selecting, copying and pasting of text; and other text/cursor controls. This method of text entering and editing for writing procedures is significantly easier for the beginner to use than previous Logo text editors. This improved method for entering and editing text again reduced the amount of time spent upon teaching how to use the system. More time could be spent teaching the programming concepts and syntax than operating system and text editing commands.

Procedures

During the semester, students in the Computers and Society course spend 50 minutes per week in the laboratory. Programming accounts for one third of the total time in the laboratory. Students received *only four one-hour sessions of instruction on the use of Logo*, each held one week apart. During the class sessions, usually two students shared a computer. Students were expected to sign up for additional time to use the computer individually. There was reasonable access to computers during the week. The laboratories were not overcrowded.

The first lab session was devoted to introducing Logo and familiarizing the students with the Apple //e keyboard. The students used the *LogoWriter* tutorial programs available on the *LogoWriter* disk. The tutorials had them practice the rudimentary Logo turtle graphic commands.

The second week laboratory included a written student guide with a list of the rudimentary turtle graphic commands and included the REPEAT command. The students were shown how to use the REPEAT command to draw an N-sided closed, symmetric figure during the laboratory session.

The third-week laboratory guide discussed how procedures can be written to call each other and was illustrated by a complete program

```
Third Week Guide        Fourth Week Guide
TO STARTUP              TO STARTUP
   HOUSE                   CG CT
   PU                      PRINT [Move the
   RT 90 FD 112              Swimmer Up or Down?]
   RT 135                  NAME READLIST "ANSWER
   PD                      IF ELSE :ANSWER = [UP]
   TREE                      [SWIMUP]
   HT                        [SWIMDOWN]
END                     END

TO HOUSE
   BOX                  TO SWIMUP
   TRIANGLE                REPEAT 20[SETSH 1 WAIT
END                          5 SETSH 2 WAIT 2 FD 20]
                        END
TO BOX
   REPEAT 4             TO SWIMDOWN
   [RT 90 FD 50]           REPEAT 20[SETSH 2 WAIT
END                          5 SETSH 1WAIT 2 BK 20]
                        END
TO TRIANGLE
   RT 45 FD 35
   RT 90 FD 35
END

TO TREE
   TRUNK
   CIRCLE
END

TO TRUNK
   FD 25
END

TO CIRCLE
   LT 90
   REPEAT 20[FD 5 RT 360/20]
END
```

Figure 1: Programming Examples from Guides

of seven procedures including a STARTUP procedure used in a three-deep nested design (Figure 1). Students were given the assignment of designing their own "scene" using a STARTUP procedure. They were told that they would not have to hand in the assignment, but the scene would be part of another program that they would be required to hand in.

STARTUP is a special LogoWriter procedure name. If the programmer types a procedure named STARTUP in the procedure mode, it can then be used as a master procedure for the rest of the program. In particular, when the file (the PAGE in LogoWriter) is selected from the table-of-contents, if it has a STARTUP procedure in it, that procedure will execute immediately. If the PAGE does not have a STARTUP procedure, the system will be initialized in the immediate mode and wait for commands to be typed.

The fourth-week guide illustrated and discussed the use of PRINT to print a message on the screen, the use of READLIST and NAME for reading the value of typed input data using a variable, and the use of an IFELSE statement to branch between procedures. The program (Figure 1) was only nested two-deep and included a number of commands illustrating the use of the SHAPES option in LogoWriter to achieve animation.

Variables were not extensively discussed; it was mentioned that a variable could be used for other than list data but students did not

The fourth week assignment was to write a program using procedures including STARTUP which would draw a "scene". They were then to use a special LogoWriter feature which allows new shapes of the turtle to be used. The shapes were to be used to give the effect of an animated figure moving through the scene. For example, a "swimmer" moving through the waves. In addition the students were expected to use interactive input and output to have the program branch between procedures that would determine different ways for the shape to move. For example, the program user types "up" or "down" to specify whether the swimmer moves up the screen or down the screen. The assignment was effectively modeled by the two programs illustrated in the laboratory guides (Figure 1). Students had a total of three laboratory sessions to develop this program.

Experimental variables

Collection of information about the students and analysis of their programs was done after the fifth week of the semester.

Background questionnaire and Logo exam. Students answered a background questionnaire including 22 questions giving information on: hours spent working on the program; previous computer experience; amount and quality of assistance from teachers, laboratory assistants, or other students; and other questions about mathematics background, GPA and other possible predictors of success (Lawson, 1985). All questions used in this analysis had a response which could be ranked numerically.

In addition they took a paper and pencil Logo exam in which they: 1. analyzed control flow in a sample program demonstrating their understanding of hierarchical procedure calls in Logo, and 2. played turtle by drawing the graphics which would be produced using a very simple Logo procedure.

One hundred students completed the questionnaire and Logo test in class, and turned in a program for the Logo laboratory assignment.

Evaluation of students' Logo programs. Based on the evaluation of the students' programs and the questionnaire, only those students whose programs successfully met the Logo laboratory assignment criteria and who had had NO PRIOR programming experience were selected as a subsample for analysis of their programs.

Out of the 100 students, 45 students successfully completed *all* of the requirements for the assignment, 26 of those students had no prior programming experience. Those students who were previously inexperienced and successful Logo programmers were selected for the program evaluation.

The evaluation of the 26 Logo programs is essentially descriptive, designed to describe how each student sub-divided the entire program into modules (procedures), organized the program into a hierarchical structure (depth), and used iteration to maximize the power of the program (multiple procedure calls and REPEAT statements).

Printed copies of the subsample students' programs were used to count: 1. non-procedure name Logo commands in the program (total atomic commands); 2. procedures defined by the student including the LogoWriter procedure, STARTUP; 3. procedure calls made in the program; 4. atomic commands in the longest procedure of the program; 5. REPEAT commands (iteration) used; 6. the maximum depth of procedure calls; and 7. the number of times an independent series of procedure calls was made to that maximum depth.

Atomic commands include standard turtle graphic commands which primarily included: FD, BK, RT, LT, HOME, PU, PD, PRINT, READLIST, REPEAT, HT, ST, IFELSE,HOME, SETSH and WAIT.

Maximum depth (#6, above) of procedure calls was determined by locating the first call for a each procedure and setting the depth at one, and following the tree of procedure calls to its maximum depth where the final procedure only consists of atomic commands. Once maximum depth was determined for a program, the number of times (#7, above) the program used independent paths of that depth was counted.

Results

A complete correlation matrix was generated comparing the values for the 22 questionnaire questions, the two scores on the Logo exam, the students' scores on their Logo laboratory program assignment, and the properties used to analyze the successful programs.

Summary data (mean, standard deviation, minimum, maximum) were calculated for available data on the questionnaire, programming assignment, the Logo quiz (both parts), and the seven properties of the successful programs.

Table 1: Correlations Discussed in Text

Variables	n	r	sig.
Number of atomic words with Number of words in longest procedure	26	.7334	.000
Number of procedures with Number of procedure calls	26	.8738	.000
Number of REPEAT commands with Number of procedure calls	26	.5650	.003
Number of procedures with Number of atomic commands	26	.5453	.004
Value of extra assistance with Number of atomic words	19	-0.3431	.150
Score on laboratory assignment with Number of extra hours worked	100	.1822	.070

Correlation data:

Interesting results from the large correlation matrix are summarized in Table 1. Almost none of the questions in the questionnaire had predictive value for performances on the laboratory assignment, the Logo quiz, or the structure of the successful Logo programs. The exception was a strong negative correlation which occurred between the length of a successful student's program (number of atomic words) and the student's estimate of the value of the assistance which they received from others in writing their program.

Certain features of the successful programs correlated highly with each other. The length of the program (number of atomic words) correlated positively with the length of the longest procedure in the program. This suggests that students wrote longer procedures when they wrote longer programs. At the same time there is a reasonably high correlation between length of program and the number of procedures. That suggests that most students are continuing to subdivide the program, and not just working with a set number of modules in mind. They went beyond the structure of the programs in the laboratory guides.

Not surprisingly, the number of procedures defined in the program was highly correlated with the number of procedure calls. Not shown here, but related to this observation, is that no successful student included a procedure in their program that was not called.

The number of REPEAT commands used is positively correlated with the number of procedure calls used. This suggests that students who appreciated the organizational utility of writing procedures with only a few sequential commands, also understood the generative power of iteration applied to a few sequential commands. These concepts may be related in the student's mind.

The above results suggest that students use those language features for control structures to which they have been introduced. The longer programs are not longer because the author wrote long, sequential command code. The longer programs are as complex in flow of control per atomic command as the shorter programs.

Table 2: Scores of Logo Programs and Exams

Variable	Mean	Std Dev	N
Logo Lab Assignment (15 possible)	10.1	4.8	100
Control Flow Exam (4 possible)	2.3	1.5	45
Play Turtle Exam (3 possible)	1.8	0.8	45

Summary data on scores:

Table 2 summarizes the scores students received on their programs and the two parts (Control Flow and Play Turtle) of the Logo quiz.

The mean score for 100 students who turned in the assignment was 10.1. A score of 12 or better means that the student had a working program which successfully implemented STARTUP, drew a scene, requested and evaluated input from the user, and used the SHAPES option to produce an animated turtle. A score of 10 would imply one or two of these properties was not working. A score higher than 12 indicates that the program was more complex than the minimal requirements; complexity was judged by what appeared on the screen. Subtracting or adding points to 12 was not done in a consistent, rigorous manner.

The Control Flow part of the quiz had a range of values from 0 to 4. A score of 3 or 4 would indicate the student traced the flow of control correctly. The mean of 2.33 for 45 students who took the quiz indicates that most students could not perform the task. This is not necessarily an accurate evaluation of their understanding of procedure calls in Logo. Students had never seen this type of task before. There was obviously serious confusion about what was expected.

The Play Turtle quiz had a range of 0 to 3. There were 3 parts to the graphic drawn by the program: a square, a triangle, and their relative positions on the screen. Scores were determined based on one point for each of the parts. A mean score of 1.84 indicates that most students could Play turtle for a while, but not for long. In particular most students correctly drew the square which they had previously drawn using Logo in the laboratory. The other commands were new.

In summary, it is difficult to evaluate students' understanding based on the program and quiz scores. More careful and detailed analysis will be necessary before being able to specify what most students understand about the syntax of atomic commands and procedure calls.

Summary data on programs:

Reading and evaluating the 26 sample programs revealed two important ways they all differed from each other: the data (scene and animation), and the design of the flow of control. Students were quite original in their scenes and animation, ranging from sailboats competing in the Americas Cup, to animated faces peering out from windows. Students did not simply copy the house and tree illustrated in the laboratory guide; only a few programs included these at all. Neither did students count the number of procedures in the example programs, and stick to that as a model. All of the sample programs showed that students generalized about the process of using procedures.

Table 3 gives the summary data on the sample students' programs. As mentioned above, the students wrote very different programs.

As suggested in the data, students readily break their code down into procedures. The mean program length (atomic commands) was 114.7, and the mean number of procedures was 10.1. This gives an estimate of mean procedure length of approximately 11 atomic commands which is a reasonable size for a procedure in most programming languages. However, individual students varied in maximum procedure

Table 3: Evaluation of Successful Logo Programs

Variable	Mean	Std Dev	N
Number of Atomic Words	114.7	60.5	26
Number of Words in Longest Procedure	43.0	28.2	26
Number of Procedures	10.1	4.5	26
Number of Procedure Calls	11.0	6.5	26
Number of REPEATS	5.8	3.5	26
Depth of Procedure Calls	3.4	0.8	26
Number of Procedures at Depth	1.6	1.4	26

length. The range of atomic commands in the longest procedure is from 9 to 117. The student who wrote the procedure with 117 atomic commands did use 11 total procedures; the remaining 10 were fairly short.

The mean length (atomic commands) of the programs ranged from 26 to 305 commands. The student who wrote the program with 305 atomic commands used a maximum depth of 5 procedure calls. The student who used a maximum depth of 6 procedure calls wrote only 84 atomic commands, well below the mean length.

Some programs had multiple calls of the same procedure as indicated by mean number of procedures (10.12) and mean number of calls (11.04). Most programs had one less procedure call than procedures. The latter is the pattern when the STARTUP program begins the program (without being "called") and the remaining procedures are called once in the program.

All IFELSE statements that students used involved evaluating a list variable which received its value from user input. Students were not told about alternative variable data types or ways of using conditional branching. Although not analyzed formally, it appears that most students only used one IFELSE command; they used it in the manner of the example in the guides. A few sample programs had more than one IFELSE, but these also were used similarly to the example program.

A summary of the results from reading and evaluating the sample student's programs indicates they have a good working knowledge of how and when to use sequencing and iteration; they also wrote their programs in good modular style. Working from only a few examples in the guides, students wrote very original procedures of reasonable length. They used the REPEAT statement effectively, demonstrating their understanding of the power of iteration. Several programs had multiple calls of the same procedure, again demonstrating those students' understanding of this source of computation power. Multiple procedure calls had not been illustrated in guides, although it was mentioned in class. Only a few students used the IFELSE statement in an original way; for the most part they simply used it the way it was illustrated in the guides.

Discussion

This report discusses the results of preliminary analysis of data collected in a computer science course for non-majors. Based on these results, we plan to evaluate our approach to teaching both computer science majors and non-computer science majors. Our evaluation depends on the objectives for the particular course.

The teaching objectives of computers and society courses continue to evolve and vary from institution to institution. Whether or not students should learn about computers by learning programming is under debate. The course given at Towson State University is structured so that students spend one-third of their class time in computer laboratories working with the four most common types of software: programming languages, spreadsheets, database management and word processing. The expectation is that students will learn enough about the rudiments of these programs and their applications that they will be able to appreciate each one's potential for solving a particular class of problems. Ideally the student will understand the central features of the software so that they will be able to know how to evaluate software and be better prepared to use it in the real world.

Does the data type of the programming language? What do students learn about the turtle graphics data in this course? Students do not perform particularly well when asked to "play" the turtle on paper. As shown by our class quiz, most students could not accurately draw the turtle graphics for a simple square and triangle. Perkins, et al (1986) makes the same observation about students learning Logo and argues that the turtle graphics domain is not as trivial and easy as is usually assumed. Perkins points out that established psychological research shows that people do not reliably perform on tests requiring them to quickly select left versus right movements. It seems reasonable to expect that most people are not skilled in performing spatial tasks requiring precise degree turns, particularly in relative geometry. It is clear that the computer does a better job of being accurate than a human.

However, turtle graphics are manageable for students whether they learn to be the turtle or not. It is clear that students enjoy meaningful success at manipulating turtles in

Logo. There is some evidence that the Logo turtle graphics experience does teach the learning programmer to understand relative geometry. Clements (1984) found that young children who wrote Logo programs were able to perform better on a pencil and paper test which had them describe how to move through a two-dimensional map. Learning to use turtle graphics or relative geometry, however, is not a sensible objective for a computers and society course. Learning about the power of control structures is an important objective for this course.

Most programming languages have the same fundamental control structures: sequencing; iteration; and branching. It is this fundamental source of computing power that is relevant to all future computer users and appreciators.

If the focus of the laboratory programming exercise is on control structures, it is reasonable to teach students to use them in the context of using the turtle graphics of Logo. Students are motivated and able to write a program after a few hours' instruction. The data (movement of the turtle) are relatively easy to manipulate, particularly because variables are not necessary. At the same time defining procedures and making procedure calls is particularly simple in the Logo language.

The ease of writing procedures in Logo has been well discussed. Pea (1985) describes the language's "procedurality" as a particularly valuable tool for learning to construct programs from small modules. Dale (1984) makes the argument that "proceduralizing" in Logo is easy to learn because procedure names are used in a manner parallel to words in natural language. Both procedure names and natural language words "become multi-purpose and can be used in different contexts." This does not argue the in this study students are programming using either a top-down or bottom-up approach. Students were not instructed regarding either strategy. The evidence is simply that Logowriter provides a programming environment where the naive student readily modularizes and organizes the program hierarchically. Our teaching research suggests that with very little teaching (remember, only four hours of instruction were given), many students are able to write programs in a well-modularized and hierarchical style.

There is very little research in the literature which analyzes teaching and student performance by evaluating the programs that the students actually write. Given the focus of teaching students the principles of flow of control in programming languages and the observations made in this study, it seems reasonable to emphasize control structures during the earliest laboratory lessons. It appears that students don't need to practice working with the turtle graphics commands for several weeks before moving on to procedures. It may be that if students spend much time writing long, sequential command programs, they will not be as willing to switch to a well modularized, hierarchically designed programming style.

In our own teaching, we are interested in continuing to teach Logo programming but with, once again, even more emphasis on structured (Bennett and Walling, 1986) programming style. We may further emphasize multiple procedure calls and more versatile applications of IF and IFELSE statements in our own instruction. Continued and more complete analysis of students, unsuccessful as well as successful, programs should help us understand the extent to which data can be deemphasized, and control flow emphasized.

The question remains, of course, whether students in a Computers and Society course will learn anything truly important about programming languages even if they understand Logo control structures. Will this understanding generalize to other languages and applications? Questions such as this are important and should be addressed with careful study of what students are learning and what we want them to learn.

References

Bennett, H. and D. Walling. 1986. Once again, structured programming: is it necessary? Computers in the Schools (2):171-178.

Clements, D.H., and D.F. Gullo. 1984. Effects of computer programming on young children's cognition. Journal of Educational Psychology 76(6): 1051-1058.

Dale, E. J. 1984. A Logo-physics project. Research/Technical (143). Educational Resources Information Center. U.S. Department of Education. Washington, D.C.

Lawson, John D. 1985. Predictors of success in the introduction to computers course for non computer science majors. Unpublished doctoral dissertation. University of Oregon. Eugene, OR.

Mawby, R., et al. 1984. Structured interviews on children's conceptions of computers. Technical Report No. 19. Center for Children and Technology. Bank Street College of Education. New York, NY.

Pea, Roy D. 1985. Logo programming and problem solving. Technical Report no. 12. Center for Children and Technology. Bank Street College of Education. New York, NY.

Perkins, D.N., F. Martin, and M. Farady. 1986. Loci of difficulty in learning to program. Technical Report 86-6. Educational Technology Center. Harvard Graduate School of Education. Cambridge, MA.

Perkins, D.N., et al. 1986. Nontrivial pursuit: the hidden complexity of elementary Logo programming. Technical Report 86-7. Educational Technology Center. Harvard Graduate School of Education. Cambridge, MA.

Dr. Sylvia A. Shafto
6236 Firefly Drive
San Jose, CA 95120
408-997-7918

Dr. John Lawson
Dept. of Computer and Information
Science
Towson State University
Towson, MD 21204
301-321-3780

Computer Applications for Transition Programs

Joseph P. DeMarsh Ph.D.
Pamela J. Zrihen M.S.
Southwest Business Industry
and Rehabilitation Association
Scottsdale, Arizona

ABSTRACT

The goal of this project was to develop new applications for computer technology to assist disable youth make the difficult transition from secondary schools to the working world. First, an Informational Clearinghouse of community educational, training and employment resources relevant to transitioning youth was constructed. Second, a client spreadsheet was developed, allowing information concerning short and long-term vocational goals to serve as selection criteria for computer searches of the Clearinghouse; thus, identifying the community resources specific to each youth. Outcome data from a 2 year pilot test in the greater Phoenix metropolitan area demonstrated the advantages of utilizing computer technology in the transition process.

BACKGROUND INFORMATION

THE NEED FOR TRANSITION SERVICES

The need to provide school-to-work transition services for disabled students, aged 16-21 years, is well documented in the unemployment literature. The interested reader is referred to Bowe, 1980; Levitan & Taggart, 1976; McKinney & Okeafor, 1984; Ognibene, 1984; Peterson, 1984; and Will, 1984 for national unemployment statistics for this target group, and Edgar et al., 1985; Bellamy, 1985; Ross, et al., 1985; and Wehman et al., 1985 for national underemployment data.

Locally (greater Phoenix area), this need for service was substantiated in the 1986 Annual Planning Document for the State of Arizona, which reveals that the highest unemployment rate for a target population in Maricopa County is accorded to the non-institutionalized handicapped population. More than 50% of this population is unemployed; that is 51,545 of the 87,180 handicapped residents are "jobless", dependent on family, community or government subsidies for economic sustenance. The subgroup of this population, by age group, with the most severe unemployment problem is handicapped youth aged 16 - 21 years. Handicapped youth number 9,272 in Maricopa county; 6,212 of these individuals stand outside the labor force.

NEEDS OF TRANSITION SPECIALISTS

In the 1983 Amendments to the Education of the Handicapped Act (EHA P.L. 98-199) Congress sought to address directly these major educational and employment transition difficulties by allocating $6.6 million annually in grants and contracts for services assisting these youth. Since that time, transition specialists have found themselves fulfilling a variety of roles including, outreach coordinator between schools and community resources, special needs counseling, student advocacy, parent education, and job placement, just to name a few (Chadsey-Rush, 1986; Cobb & Phelps, 1983). One activity which has been consistently identified as primary to most successful transition programs however, is establishing linkages between post secondary training resources and the graduating disabled student (DeMarsh, 1986; Phelps, et al., 1986).

With only 29% of disabled youth receiving vocational training during their secondary school education, many youth find themselves in need of additional post secondary vocational and other training to be truly viable in the competitive job market. As the unemployment figures for this population indicate; however, the majority of disabled graduates never attain the skills necessary to gain employment.

That educational/training resources exist is not the problem; as over 400 such services were identified in the greater Phoenix area alone. Transition counselors and special educators consistently identify the problem more as the lack of an organized and reliable information source about relevant community educational/training and employment resources. Disabled youth as well as their families and teachers are unfamiliar with the various resources available, limiting the post secondary options for training and employment considered upon graduation. Additionally, community resources' status often change from year to year (eg., services offered, eligibility requirements, location, fees, etc.) which further exacerbates this information gap.

PREVIOUS COMPUTER APPLICATIONS IN SPECIAL EDUCATING AND REHABILITATION

Several fine reviews are currently available which describe "state of the are" in computer applications for the special education and rehabilitation fields. Because such a review is beyond the scope of the present paper, the interested reader is referred to Nave et al., 1983; Backer & Trotter, 1986; Foulds, 1982; Favine, 1980;Blaszczyk, 1981; and Bennett, 1982. These reviews indicate that the majority of computer applications in the rehabilitation and special education fields has been primarily dedicated to:

1. Providing adaptive or functional aids for specific handicapping conditions such as speech synthesis (Barker, 1982; Batie, 1981; Deane, 1982; Semancik & Curtis, 1982;Sargent & Nyerges, 1979; Watson, 1979) the electronic mailbox (Cerf, 1978) touch sensitive monitors (Allard & Thorkildsen, 1981; Jung, 1980; Thorkildsen, et al., 1979) telecommunications (Aeschleman & Tawney, 1978; Johansen et al., 1981; Nocerino et al., 1978) functional aids (Aylor et al., 1981; Administrative, Training and Publication Staff, 1982; Rahimi, 1981; Vanderheiden, 1981; 1982) and other prosthetic applications (Goldberg, 1979).

2. Providing computer assisted instruction for a variety of handicapping conditions and educational and performance levels such as the blind (Ballenger, 1979; Evans & Simpkins, 1972; Hoffmeyer, 1980) deaf (Arcanin & Zawolkow, 1980; Culbertson, 1974; Dolman, 1980; Watson, 1979) and mentally retarded (Allard & Thorkildsen, 1981; Bitzer, 1979; Brebner & Hallworth, 1977; Lally, 1981; Thomas, 1981; Thorkildsen, 1982; Thorkildsen & Allard, 1980) among many others.

3. Providing management information strategies for institutions (Bower, 1975; Fleming & Vance, 1974; McGuire, 1981) sheltered workshops (Boles, 1980; Cimler & Henderson, 1979) vocational rehabilitation centers (Administration, Training and Publication Staff, 1982; Crawford, 1980; Stern & Rubin, 1979) data storage and manipulation as in job banks (McCroskey, 1986; Houghton, 1986; Soforenko, 1974) and information banks of materials on microcomputers for disabled individuals (Apple, 1986).

4. Providing computerized client assessment instruments (Cartwright & Cartwright, 1973; Hall, 1974; Hall et al., 1974; Hasselbring & Crossland, 1981, Lerner & Schuyler, 1974; Macleod & Overheu, 1977; McGuire, 1981; Wilson & Fox, 1982).

It is apparent that computers are utilized across a wide continuum of applications within the special education and rehabilitation fields. However, there has been little work done to date in utilizing computers to assist transition specialists with their difficult task of establishing linkages between the graduating disabled students and appropriate community resources. Given the developments of computer technology in the areas described above, it seems reasonable to expect that computers can also make significant contribution to the transition field.

COMPUTERS IN TRANSITION

In this project, computer applications for transition specialists took the form of an Information Clearinghouse of community resource, and a client spreadsheet that could be used in conjunction with the Clearinghouse to identify resources relevant to the long and short-term vocational goals of each youth.

INFORMATION CLEARINGHOUSE

To meet the needs of transition specialists for a current and centralized information source specific to the field, a computerized Information Clearinghouse of community educational, training and employment resources was designed. The Clearinghouse has been constructed on a variety of commercially available IBM compatible data-base manager software programs. It is designed around a catalog format similar to ones developed by the Prentke Romich Company (which lists materials available on microcomputers for disabled individuals) and the Trace Research and Development Center at the University of Wisconsin (which is a software registry and listing of programs adapted for rehabilitation applications).

The initial step in developing the Clearinghouse was the collection and analysis of all community resources relevant to transitioning youth. Resources included private for profit, private non-profit, government agencies, educational and other service providers. The focus was on analyzing and coding each resource according to the following variables;

- Target group served
- Target area served
- Age group served
- Fee schedule for services
- Intake procedures
- Contact person
- Programs or specific services offered
- Eligibility requirements
- Funding sources
- Phone number
- Address

Once coded, the information was entered into the data-base manager with on record per resource (see figure 1). The file containing all community resource records is called the Clearinghouse. Once the Clearinghouse was completed, the input of specific client data (age, disability, etc.) and client transitional needs (eg., work experience program, vocational evaluations, transportation, financial assistance, etc.) provides immediate access to resource information tailored to each client´s needs. Based on client input data, the program searches the community resource file and outputs a listing of agencies and contact persons that may be potential referral sources. Project staff can further screen these sources for eligibility and intake criteria.

INSERT FIGURE 1 HERE

The benefits of the Clearinghouse include:

1. Instant access to all community education, training and employment resources relevant to transitioning secondary disable youth.

2. An information source that can be easily updated and modified as the status of corresponding community resources change.

3. Search, order, and sort commands which allow each client´s short and long-term vocational goals to serve as selection criteria.

CLIENT SPREADSHEET

Client spreadsheets were developed to assist with both data recording and storage, and to interface with the Information Clearinghouse. Except for changes in field content and length, these spreadsheets are similar to many currently available models such as Brown (1982), Birchwell (1976), Lehrer (1978), Powers (1976) and others (see figure 2).

INSERT FIGURE 2 HERE

The client spreadsheet fills a variety of roles including: 1) an information management and client record filing system that is easily updated and stored; 2) a source file from which a variety of program forms can be generated (eg., Individual Transition Plans); 3) information concerning vocational goals which can then be utilized as selection criteria for Clearinghouse searches.

PILOT TEST RESULTS

The following section provides results from Project Bridge, a Federally Funded Research and Demonstration Program that pilot tested the aforementioned computer applications for the transition field. One hundred fifty youth were enrolled from five participating secondary schools in the Greater Phoenix Metropolitan Area on a first come basis. This sample contains a fairly representative sampling of disabilities (see figure 3).

INSERT FIGURE 3

PLACEMENTS

The goal of Project Bridge was to link disabled youth graduating from secondary schools with community training/educational resources and ultimately, with jobs in competitive or non-competitive employment settings. The philosophy grounding this effort maintains that many disable individuals exiting the secondary public school system may require additional training and or education to be truly viable in the competitive job market. Thus, by establishing linkages with community training/educational resources, the youth can then proceed with job placement activities once the necessary skills have been acquired. This section will describe placement activities at both levels; 1)training/educational sites, and 2) employment sites.

A. Training and Educational Placements. The number of youth requiring additional training and/or educational experiences prior to job placement closely approximates the number originally projected. As of 10/1/86, 139 youth were linked to community resources prior to job placement. Table 1 presents the actual numbers of these placements along with a typology of the resources utilized. An Examination of Table 1 reveals a large reliance upon job skills training, work experience activities, and specific vocational training; a finding consistent with clients' self-reported barriers to employment. Also, of all training placed clients, 35% (N = 45) were enrolled in a second program, and 7% (N = 9) went on to a third training placement. While these double and triple placements do not alter the total number of Project Bridge clients who received post secondary training programs (N = 139) it does demonstrate that project staff has actually developed 195 successful linkages with these post secondary resources to date.

INSERT TABLE 1 HERE

B. Job Placements. The total number of Project Bridge participants who are job placed (as of 10/1/86), the type of employment site and the mean hourly wage from those placements is presented in Table 2. While the individual business and organizations comprising the "Competitive Employment" category are too numerous to list, they continue to represent the typologies presented in the project's second annual report, including; food preparation, janitorial, clerical, maintenance, but with the addition of several Government and National Guard positions.

INSERT TABLE 2 HERE

Table 2 demonstrates that 58% of the entire Project Bridge sample of 150 youth have been job placed. Comparing these placement figures to the 60 - 80% national unemployment rate for the same age youth, hints at the outcome effectiveness of this transition program.

Table 2 also indicates that of the 87 youth who were job placed, 14% (N = 15) were placed at a second employment site, and 3% (N = 2) at a third. Again, while the actual number of Project Bridge clients who were job placed remains at 87, these double and triple placements of selected clients means project staff actually secured 104 job placements to date.

Of equal significance to the total placement figures; however, is data which evaluates how useful the Project Bridge Model is in managing the transition process in its entirety. It was anticipated that the majority of graduating disabled youth would not proceed directly to job placements, but would first require additional training/educational experiences. The Bridge Model was designed in part to specifically manage and coordinate this transition process to increase the probability of subsequent employment.

C. Training-to-Employment Placements. Of the 139 clients who were linked to pre-employment training and/or educational resources, 57% (N = 74) have been job placed (see figure 4).

INSERT FIGURE 4 HERE

Although this figure meets the rate projected the Bridge proposal, it is lower, in a relative sense, when compared to the over-all placement rate this project has attained to date. Several factors may be identified that contribute to this statistic. First, many youth enrolled in training sites are still involved in those courses. Obviously, job placement cannot occur until after graduation. Second, 14 youth who were linked with Summer Youth Programs (work experience and skills training) were not graduating during the 85/86 school year, and returned to their respective schools following their summer training programs. Third, a number of youth (N = 13) were job placed as no current openings in relevant community resource sites were available. For example, the Arizona State Department of Economic Securities (Developmental Disabilities and Vocational Rehabilitation) could provide needed sheltered workshop or vocational skills training only after a 1 - 3 year waiting period. Rather than have the youth sit idle during this waiting period, they were job placed.

CONCLUSION

Although placement statistics are encouraging, utilizing an Information Clearinghouse and client spreadsheet does not address all the issues involved in securing successful post-graduation training and/or job placements for disabled youth. Transition specialists will always be challenged to attain; 1) an appropriate "goodness of fit" between client and placement, 2) effective follow-up and support services, 3) increases in work adjustment and work establishment behaviors, and other necessary activities. An Information Clearinghouse does assist the transition specialist, the graduating student and parents, with identifying what resources are available with in the community that address the specific requirements of a particular student. Further, an automated client spreadsheet can generate several necessary program forms and information in a time and cost-efficient manner. Although not sufficient , the aforementioned benefits of computer applications do assist with necessary steps in successful transition programs.

Joseph P. DeMarsh Ph.D.
4410 N. Saddlebag Trail
Scottsdale, Arizona 85251

(602) 949-0135

Developing a Knowledge-based Simulation for Microcomputers

Thomas McFarland - Lewis Clark State College
Mary McFarland
Stephen Ragan
Diana Kottkey

ABSTRACT

This presentation describes the concurrent development of an arterial blood gas expert system and knowledge-based (KB) simulation. The KB simulation was used to teach nursing students the concepts of arterial blood gas interpretations. By combining the processes and tools of both knowledge engineering and instructional system design, the team was able to develop both an expert system which is capable of interpreting a wide range of cases and a simulation which provides "why", "show" and "trace" feedback to the student during training with a carefully sequenced range of cases.

CONCURRENT DEVELOPMENT PROCESS

The concurrent development undertaken at Lewis Clark State College involved combining the processes of knowledge engineering and instructional systems design. Hayes-Roth, Waterman, and Lenat (1983) describe the "evolution" of an expert system" to include the following stages: identification, conceptualization, formalization, implementation, and testing. The concurrent development process expands these stages of knowledge engineering to include components of instructional systems design related to KB simulations.

This concurrent development process can be implemented on low-cost microcomputers using expert system development tools which are commercially available. Concurrent development may be accomplished by interdisciplinary teams of professionals. These teams may include members with knowledge of the problems area (domain experts) and members with a knowledge of instructional system design and training in the use of expert system development tools (instructional designer/knowledge engineers).

Four faculty members from Lewis Clark State College developed concurrently a prototype expert system and KB simulations to guide and teach nursing students in the interpretation of arterial blood gas results. Prior to prototype development, the participating faculty members had been trained in the development of microcomputer-consulting systems using expert system development tools including M.1 from Teknowledge, Inc. As a part of a Northwest Area Foundation Grant, these two education faculty members used their previously acquired instructional design skills and recently acquired expert system development skills to work with two nursing faculty members in the clarification of a knowledge base and the concurrent development of an expert system and KB simulation involving arterial blood gas interpretation. The purpose of this expert system and KB simulation is to allow nursing students to apply the rules of arterial blood gas interpretation to a range of cases both with and without the consultation of the blood gas interpretation program.

REVIEW OF RESEARCH

The term knowledge-based (KB) simulation is a recent descriptor used by the Society for Computer Simulation to describe applications of artificial intelligence to simulation research and applications. Our work on KB simulations derives from the intelligent tutoring systems (ITS) work of Clancey (1986) at the Stanford Knowledge Systems Laboratory and the "if-then" template model of Ferrara, Prater and Baer (1987) at the Utah State University--Artificial Intelligence Research and Development Unit. This category of ITS has been labelled as "articulate expert systems" (Kearsley, 1987).

The Stanford Knowledge Systems Laboratory has explored the feasibility of using knowledge-based expert systems as a component in a computer-based teaching system. GUIDON, a computer program developed as a separate teaching knowledge base, was used to train medical students to acquire the skills of MYCIN, an expert consultant system with diagnostic knowledge related to infections and medications. Recent research on NEOMYCIN has explored the feasibility of separating teaching knowledge from medical knowledge so that each can be reused and adapted independently (Clancey, 1986).

The Artificial Intelligence Research & Development Unit at Utah State University is exploring the feasibility of implementing consultant systems on low-cost microcomputer systems. One project has involved the development of CLASS.LD, an expert system which uses rules for classifying students as learning disabled to consult with teachers and administrators regarding the appropriateness of classification. LD Trainer, a simulation program, uses this consulting knowledge base to teach learning disabilities classification procedures to teachers or administrators (Ferrara, Prater & Baer, 1987). An additional project of this Unit has been to train selected faculty in education on the development of expert systems for use in education. Two education faculty members participating in the project received summer training in 1985 at this Unit on the development of expert systems focusing on the rule-based expert system development tool, M.1 from Teknowledge, Inc.

PROBLEMS OF TECHNOLOGY TRANSFER

Artificial intelligence (AI) research in education has primarily focused upon intelligent tutoring (ITS) (Barr & Clancey, 1982; Sleeman & Brown, 1982; Kearsley, 1987). This research has occurred in a limited number of prestigious institutions of higher education which usually have departments of AI and receive funding from the Department of Defense. As indicated by the ITS label, tutoring has been the focus of this experimental research. To this point in time, these research efforts have required high-cost LISP machines and significant development time. Consequently, this research has been limited to only a few content areas.

In contrast, computer-based instruction research in education has demanded that the effects of low-cost microcomputers be evaluated for all content and application areas. Tutoring is only one area of interest--others being simulation, drill & practice, and tutee modes. Many researchers in educational technology have focused on instruction design and effectiveness of computer-based instruction.

The potential effects of collaborative efforts between educational technology and AI researchers has been a topic of a recent issue of *Educational Technology* (Ragan & McFarland, 1987). One potential area of collaboration seems to be the use of expert systems as a component of a computer-based instruction systems. To develop such a system, the knowledge base would need to be clarified concurrently with instructional design. Since the expert system would contain the knowledge base in the form of rules, the student interacting with the KB simulation would be able to ask "why" or to "trace" the rules being used by the expert system during the simulation. Since several expert system development tools are available for microcomputers, the development of KB simulations appear feasible for cost-effective delivery. Another proposed advantage is that the developers of expert system and KB simulations need not receive intensive training in knowledge engineering.

The implementation of this process in the development of an expert system and a KB simulation is described using examples from the arterial blood gas interpretation project at Lewis Clark State College.

STAGE 1--IDENTIFICATION

Expert System

Needs Assessment

The team assessed the needs of nurses and nursing students. One priority established during the needs assessment process was for a consulting system that could guide nursing students in the interpretation of arterial blood gas reports.

Problem Selection

Utilizing domain and task selection criteria, the team decided that the problem met the minimum requirement for a structured selection or rule-based expert system because the solutions were enumerable, facts were stable during consultation, and the task could be competently performed by an "expert" nurse.

KB Simulation

Purpose Clarification

The purpose of the KB simulation was to provide an instructional delivery system that would teach nursing student to interpret arterial blood gas reports with a range of cases from simple to complex.

Writing Instructional Objectives

All instructional programs require specific objectives to clarify what the student will learn. (For example, one objective for this KB simulation is that the nursing students will examine a range of arterial blood gas reports and make accurate interpretation as to whether the results show an acidosis or alkolosis).

STAGE 2--CONCEPTUALIZATION

Expert System

Find Core Concepts

Core concepts were found by using both "compiled" sources--articles and textbooks--and "expert" nurses. The "expert" nurses determined the core concepts of arterial blood gas interpretations by analyzing design cases previously used in the testing of nursing students.

Find Critical Attributes

To find critical attributes, the team listed possible interpretations and identified critical attributes for each core concept. (For example, the critical attributes of the core concepts pH--acid base, HCO3--bicarbonate, and pCO2--partial pressure carbon dioxide were required to make correct interpretation for all design cases).

KB Simulation

Conceptual Analysis

The team agreed that instruction would focus on clearly communicating information about the three core concepts determined by "expert" nurses. In addition, supplementary information regarding acidosis, alkolosis, metabolic, repiratory, compensated, and uncompensated is provided for individuals using the program.

Task Analysis

The tasks required for correct interpretations were determined to be data-related decisions focusing on the three core concepts (For example if pH is less than 7.40 then the interpretation is acidosis). A task hierarchy of data-related decisions was established as the prototype instructional sequence.

STAGE 3--FORMALIZATION

Expert System

Design Control Structures

The control structures determine the order of questioning during the consultation. M.1 uses backward chaining as the rule-based expert system seeks to find whether the first "interpretation" encountered is true. M.1 allows for the modification of the inference process with commands such as "initialdata", "presuppositions", and "whenfound".

Write Questions

The team developed initial questions for the consulting process. During prototype development, the questions were modified to provide better questions with guidelines on data-related decisions.

STAGE 4--IMPLEMENTATION

Expert System

Formulate Rules

If-then rules were written for each interpretation using the core concepts and critical attributes. Most rules involved data-related information on the core concepts. The following M.1 rule illustrates data-related information:

if pCO2 = C and C = 40.0
then pCO2 = absolute-normal.

Write Displays

Displays were used to provide supplementary information to the user during consultation. Information was provided on core concepts, interpretations, critical attributes and interpretations.

KB Simulation

Select Instructional Procedures

Fact systems were selected to initially teach concepts and relationships. Fact systems are instructional procedures which demonstrate the visual-spatial relationships between seven to twelve concepts while presenting a minimum of incidental information (Engelmann & Carnine, 1982). A case method simulation follows the initial instruction.

Select Instructional Media

The media compatible with fact systems is the overhead projector and transparencies to convey the visual-spatial relationships. Extended teaching for the arterial blood gas interpretation project used IBM or compatible microcomputers.

KB Simulation

Direct Instruction

The team implemented direct instruction using overhead transparencies to communicate the core concepts and critical attributes during initial teaching. Additional transparencies were used to communicate the relationship between the critical attribute data-decision points of each core concept.

Simulation with Teaching Cases

Teaching cases requiring only discrimination of one concept were used for initial instruction. The teaching cases for the last interpretations involved multiple and subtle discriminations required of "expert" nurses. The program allowed nursing students to simulate cases of their own choosing.

STAGE 5--TESTING

Expert System

Validate Rules

Test cases were maintained by the team to allow for validation of the rules and checking to monitor the correctness of the program. At present, the test cases reflect the knowledge base required of an "expert" nurse. An arterial blood gas interpretation program written for a laboratory technician or doctor might include different report forms or cases.

Redesign

The team is seeking new cases which the present program cannot adequately answer in order to provide new directions for revision and to understand the limits of the program. The team will continue to revise the rules, questions, and displays as needed.

KB Simulation

Formative Evaluation

The usefulness and effectiveness of the KB simulation was explored with nursing students. All nursing students received direct instruction on arterial blood gas interpretations and one-half of the students interacted with the KB simulations using the appropriate range of cases.

Summative Evaluation

The purpose of summative evaluation is to determine how well the product works in comparison to other delivery methods. A controlled-research experiment with the revised product will be established using an appropriate group design.

DISCUSSION

Although a number of intriguing issues and questions have emerged during this concurrent development process, we have selected two for the conclusion of this paper. The first question involves the level of competency required for knowledge engineering. The educators acting as knowledge engineers had received only two weeks of structured training on the expert system tools and knowledge engineering process. Both individuals had read widely on the process of knowledge engineering, however, neither has prior experience in the knowledge engineering process. Several articles have suggested that a background in applied behavior analysis (psychology) and instructional systems design (education) might be useful to knowledge engineers. Although we would have preferred to work with an experienced knowledge engineer with a computer science background, we were able to develop a small rule-based expert system without this support. However, it should be noted that the expert system development tool, M.1, greatly influenced the selection of the arterial blood gas interpretation problem. Obviously, it would have been significantly better to first select the problem, and then to seek an appropriate tool for it's solution.

The second involves the usefulness of the concurrent development process. Given the background of our development team, we have found this process to be useful for the development of the arterial blood gas interpretation program. As indicated, we are presently using formative evaluation procedures to determine the effectiveness of this program. We have plans to use summative evaluation procedures during this school year. The team is satisfied with both the concurrent development process and the initial products. Both nurse "experts" indicate that the clarity of their teaching has improved as a result of the knowledge clarification process required during concurrent development. We are presently developing other prototype expert systems and KB simulations for nursing and education with this concurrent development process.

REFERENCES

Barr, W. & Clancey, W.J. (Eds.)1983, *Handbook of Artificial Intelligence- Vol. III*. Los Altos, CA: William Kaufman, Inc.

Clancey, W.J. 1986. "From GUIDON to NEOMYCIN and HERACLES in Twenty Short Lessons: ORN Final Report 1979-1985." *AI Magazine* 7(3), 40-60.

Engelmann, S. & Carnine, D. 1982, *Theory of Instruction--Principles and Applications*. New York, NY: Irvington.

Ferrara, J.M., Prater, M.A. & Baer, R. "Using an Expert System for Complex Conceptual Training." *Educational Technology* 27(5), 43-46.

Hayes-Roth, F., Waterman, D.A. & Lenat, D.B. 1983, *Building Expert Systems*. Reading, MA: Addison-Wesley.

Kearsley, G. 1987, *Artificial Intelligence and Instruction--Application and Methods*. Reading, MA: Addison-Wesley.

Ragan, S. & McFarland, T.D. "Applications of Expert Systems in Education: A Technology for Decision-Makers." *Educational Technology* 27(5), 33-36.

Sleeman, D. & Brown, J.S. (Eds.) 1982. *Intelligent Tutoring Systems*. London: Academic Press.

Dr.Thomas McFarland
Spalding Hall
Lewis Clark State College
Lewiston, Idaho 83501
208-799-2338

An Evaluation System in Dental Hygiene Clinical Education

Michael W. Sachon, Old Dominion University

Patricia-Damon Johnson, Old Dominion University

Abstract

A criterion referenced clinical evaluation system has been developed at the School of Dental Hygiene, Old Dominion University, Norfolk, Virginia. The system uses a computer for systematizing and managing clinical evaluation data according to programmed criteria and standards. It provides feedback and summative performance results to students through each of four semesters of clinical coursework in Dental Hygiene. Clinical instructors assess student performance using a custom designed optical scanning form. Computer programs have been developed to validate data read from the forms, to reject forms which are incomplete or contain errors, to update a clinical data base, and to generate progress reports. The system also provides a cummulative summary of each clinical instructors evaluations of students which is used for the standardization of faculty evaluation-skills.

AN EVALUATION SYSTEM IN DENTAL HYGIENE CLINICAL EDUCATION

Purpose

The purpose of this paper is to provide information regarding the design and implementation of a comprehensive and functional computer-managed evaluation system in dental hygiene education. Well-defined instructional criteria, objectivity in evaluation, and specific feedback on performance are desirable goals of clinical education. The student evaluation process must be orderly and systematic to accommodate these goals. The system uses a computer for systematizing evaluation data according to programmed criteria and emits feedback to students on their performance. Development of the computer managed clinical evaluation system has made the process of evaluation for faculty and students more efficient and accurate by providing the following: 1) criterion-referenced process and product evaluation standards; 2) machine processing of evaluation forms; 3) data processing and generation of student clinical progress reports; 4) efficient storage of clinical evaluation information; 5) documentation of patient statistics; 6) record keeping for accreditation purposes; 7) quantity and quality data of student progress and 8) comparison of the reliability of instructor ratings for standardization purposes.

Introduction

The computer managed clinical evaluation system has been developed by a collaborative effort between the faculty of the School of Dental Hygiene and the Staff of the Computer Based Learning Laboratory at Old Dominion University, Norfolk, Virginia. A two year time period was needed to design and develop the system with the initial implementation in 1983. After a two year pilot test (1983-1985), a redesign was implemented reflecting 1) changes in the evaluation criteria and standards, 2) changes in hardware and 3) changes in the integration of the application programs with a commercial data base management system.

The evaluation of student performance in clinical courses is an attempt at the objective validation of clinical progress. Instructional decisions must be made on which activities are to be evaluated, how achievement in those activities is to be assessed, and how difference in achievement may be discriminated. The relative value of each activity must be decided in terms of importance and difficulty to the course. Along with these instructional decisions, a mechanism must be established to collect data, store the data in a data base and process the data into reports which provide instructional feedback to students and course management information to faculty. In dental hygiene clinical courses, student data consists of patient load requirements, radiographic requirements, and clinical services requirements. Statistics must also be kept for ongoing accreditation of the program.

Students are evaluated in the clinic by clinical faculty who record evaluation data on a custom designed optical scanner form. A computer program processes the data read from the forms prior to insertion into a data base. Forms with missing or invalid data are not added to the data base, in which case faculty correct and resubmit the evaluation form for processing. The data base is managed using a commercial data base management system. Applications programs written in FORTRAN communicate with the data base management system to load records into the data base and generate cumulative performance reports. The commercial data base query language is used to update records which have valid entries but are erroneous, such as when a wrong grade is assigned to a student activity.

Description of the System

Data Collection Procedures

The Dental Hygiene Clinical Evaluation System is comprised of five components with pre- established performance criteria and standards for each component. The five components include Assessment, Planning, Implementation, Evaluation and Management. These components of the system were selected using the philosophy and theoretical of the "Nursing Process." (Yura and Walsh, 1983) The process is the tool and/or methodology which health care professionals use to arrive at decisions and predict consequences in patient health care delivery. Using the process is a means of evaluating quality of health care delivery and assures accountability and responsibility to the patient (Nursing Theories Conference Group, 1980). For each component, there are various related clinical skill areas. The clinical skills of dental hygiene students are evaluated for each patient according to the pre-established performance criteria.
Students and clinical instructors receive written criteria for each component and are aware of the evaluation process prior to the patient appointments. Students receive a mastery or non-mastery rating for the clinical skill area based on their performance of the established criteria. Clinical grades are determined by the percentage of time a student masters each clinical skill area times the weights assigned to each component. An optical scanner evaluation form is used in the clinical setting to collect student performance data for each patient visit.(See Evaluation Form) The completed form is used to provide immediate student feedback and retrievable data for analysis The evaluation form is filled out by students and instructors. The students are responsible for recording patient and appointment data and the instructors are responsible for assignment of grades.

At the end of each clinic session, instructors sign and check the forms for completion and deposit the forms in a designated area. The evaluation forms are processed at the Computer Based Learning Laboratory. (See flow chart for step by step procedures of Computer Managed Evaluation System)

Computer Programs

The present implementation of the Clinic Evaluation System is on an IBM 4381 computer running the CMS operating system. The evaluation forms are processed by an NCS optical scanner yielding a raw data file corresponding to the fields marked on the form.

Program BRUSH is a pre-processor that reads the raw data from the optical scanner data file and compares each data field from the form with lists of valid values previously stored by the BRUSH program. Field values which are not complete or do not match a pre-defined value cause the BRUSH program to output that record to a Conflict file. Preventing the majority of erroneous data from being stored in the clinic data base is a desirable effect since data base updates are performed interactively and are time consuming. The evaluation forms corresponding to the conflict records are corrected and re-submitted.

BRUSH compares the student number against the students registered in the course. The program verifies that patient data has been recorded and insures that all student evaluation fields have been completed. BRUSH also translates the raw optical scanner data into an alternate form so that the data base is loaded with values meaningful to the user. Successful records are output to an intermediate data file which is used to load a clinic data base.

The data base is loaded using an application program which interfaces to a commercial data base management system, IBM's System Query Language/Data System (SQL/DS). The application program, DHLOAD, reads the output file from program BRUSH, one record at a time, and inserts the record into an SQL/DS table.

The direct linkage between the FORTRAN applications programs and the SQL/DS managed tables is accomplished by the SQL system. The applications programmer embeds SQL commands and variables in the source code of the application program. An SQL preprocessor reads the programmers source code and replaces the SQL commands with FORTRAN code to implement the commands. The proprocessor outputs a converted source file which is ready for compilation, linking, and execution. (IBM,1984)

DHREPORT is the application program that generates the student reports from the SQL data base. As with the DHLOAD program, communication with the SQL tables is performed using embedded commands in the source program. SQL performs the selection and the sorting of the data records to be read from SQL tables. The sorted records are read from the SQL table sequentially by the program which prepares cumulative data for the student reports. DHREPORT allows individual student reports to be generated or an entire course can be processed from a file of students.

DHREPORT is used for four different clinical courses. The differences in the performance criteria among the courses are in the number of times each clinical task must be performed and the relative weight of each task to the course grade. This information is kept in a data file so that modifications can be made to the grade determining variables without making any changes to the source code. This adaptability is important to the clinical faculty so that as changes in the emphasis of courses occur they do not obsolete the system. It is important to the computer support staff that the system can change without modifying the source code.

Updates to the Clinical Data Base are performed using Interactive SQL (ISQL). A modification form was designed so that clinical faculty can specify information to identify the record to be changed and any field values in error.

Use of the Reports

Dental Hygiene students receive a clinic progress report every two weeks to update and document their performance skills (see Progress Report). The clinical administrator and each of the students clinical instructors receive a copy of the report. The reports are a cumulative record of the student's performance. Student and faculty are required to meet several times per semester for a clinical grade conference to discuss the reports and student's progress. The report provides the student with the following types of information: 1) percentage grades for each skill area, 2) tally of clinic requirements, and 3) accreditation data on type of patients treated.

The first section of the report provides the following information on grades:

1) The number of times a clinical skill area has been attempted.

2) The number of times a clinical skill area has been mastered or non-mastered with the appropriate percentage

3) The percentage weight for each clinical skill area.

4) The grade for each clinical skill area based on the percentage weight and the final grade.

The second type of information on the progress report provides the students with a cumulative tally of the number of clinical requirements completed at the time of the report. This information includes:

1) The number of radiographs completed.

2) The number of comprehensive services completed.

3) The number of patients completed vs. incompleted.

The third type of information on the report provides the student data on the type of patient treated by age with regard to classification. This information is also used for accreditation summary reports. The information from the reports provides students with specific feedback of their clinical strengths and weaknesses and allows them to make decisions on future clinical performance goals.

In addition to the progress reports, students receive two summary reports which are generated from the data base.

The summary reports allow the student to cross reference patient data and grades with the evaluation forms for validation purposes (see Summary Reports). For faculty standardization purposes, faculty receive a report twice a semester (midsemester and final week). The report provides a cumulative summary of the clinical instructor evaluations of each skill area for their assigned students. This report is used by the faculty for self-assessment in their evaluations of each clinical skill area. Additionally, this information assists the clinical administrator in the identification of faculty who evaluate in a strong or weak manner. A comparison of the reliability of instructor ratings is used for standardization purposes (see Faculty Evaluation Reports).

DISCUSSION/SUMMARY

Dental hygiene educators are continually faced with the problem of accurately and objectively evaluating student's clinical skills and performance. Considering the time spent and the frustration often experienced, the grading aspect of the dental hygiene clinical experience can be a major deterrent in the teaching process (Christiansen, 1978). Nevertheless, evaluative problems must be confronted and solved in order to facilitate the educational process. With these concerns, the faculty and computer staff collaboratively developed a computer managed criteria-based evaluation system with the following considerations and objectives: documentation, adaptation, measurement and evaluation (Biller, 1980).

Since the inception of the computer managed evaluation system, the programs have been assessed annually and modifications have been made by both faculty and computer staff. The current SQL-based system has been implemented since 1986. This continual re-evaluation has produced a successful computer managed systematic approach to clinical evaluation. The strength of the evaluation system is indicated by its acceptance and use by the clinical faculty and students. The information gives students cumulative feedback of their performance on each clinical skill area so they and clinic faculty can work toward remediation. Clinical faculty receive feedback on their assessments of each specific clinical skill area which they use to improve their clinical teaching and evaluation skills. Students and faculty are more at ease during the evaluation process since pre-established criteria have been determined and students receive written feedback during the learning process. Faculty are more standardized since the system decreases the amount of inconsistent grading practices. These factors have helped to eliminate confrontation between faculty and students and, most importantly, subjectivity in grading has been reduced.

The efficiency with which the system operates extends to the computer support staff. The system requires minimal ongoing computer staff support, only 1-2 hours per month to support two clinical courses each semester. Four different courses are served by the system, which is adaptive to changes in emphasis and requirements for the clinical courses. Student task requirements and grading specifications can be modified without changes to the programs. In summary, the development of a computer managed criteria-based evaluation system has proved to be an invaluable tool in the clinical dental hygiene teaching/learning process.

Bibliography

1. Biller, Irene R. and Kerber, Paul. Computerized Clinical Evaluation in Dentistry. Part I: Rationale, Design and Construction of Instrument. *Educational Directions*, Vol. 5, No. 1, 1980.

2. Biller, Irene R. and Kerber, Paul. Computerized Clinical Evaluation in Dentistry. Part II: Design, Implementation and Use of the Computer Program. *Educational Directions*. Vol. 5, No. 2, 1980.

3. Christiansen, Mark G. and Burgess, Judith S. Automating and Standardizing Clinical Evaluation. *Educational Directions*, Vol. 2, No. 4, 1978.

4. Nursing Theories Conference Group. *Nursing Theories: The Base for Professional Nursing Practice*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1980.

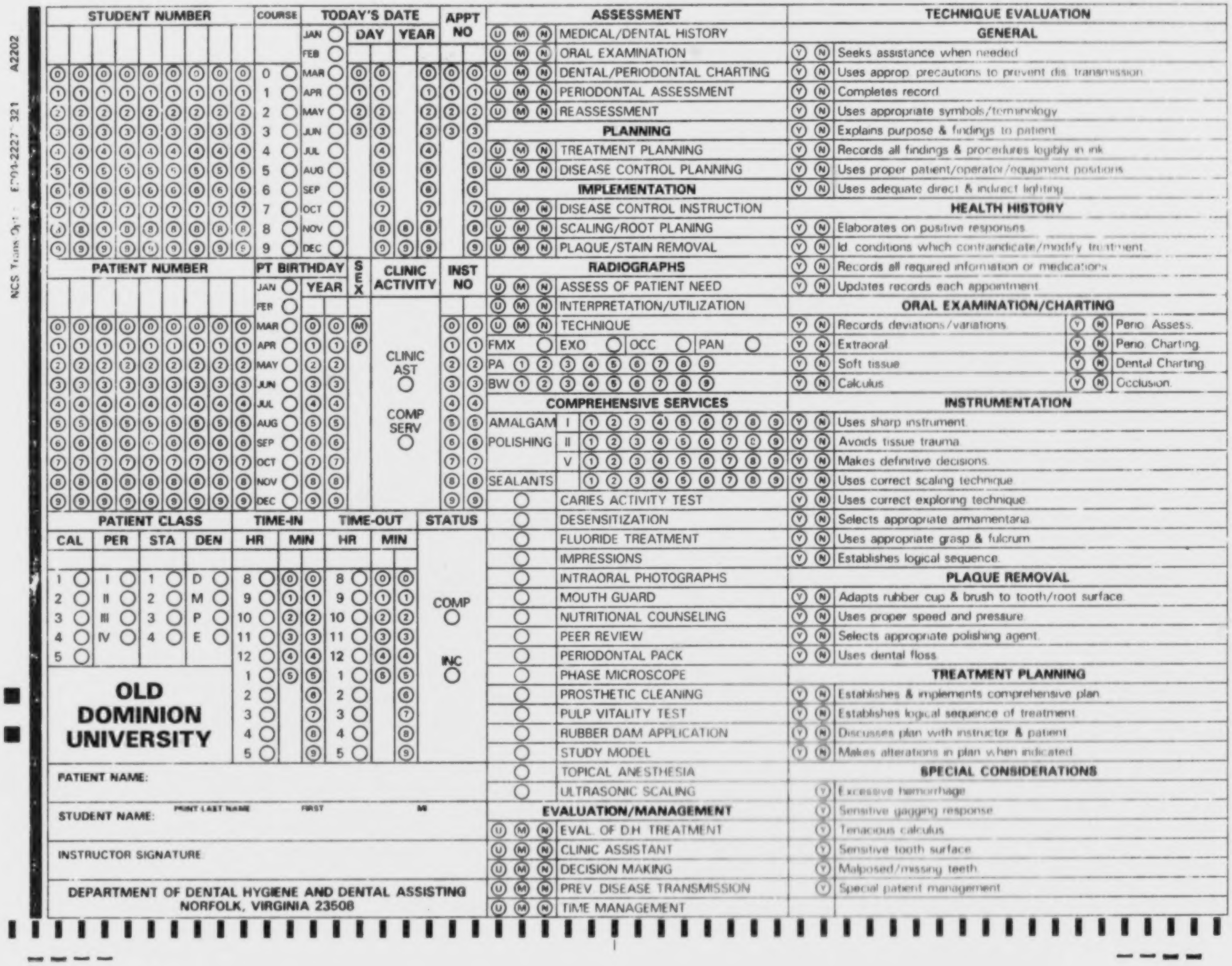
5. Saroff, Stephen A. Computer-Assisted Evaluation in a Clinical Setting. *Journal of Dental Education*. Vol. 39, No. 7, 1975.

6. Scruggs, Rebecca R. and George, Mary C. Faculty Calibration in Clinical Education: A Review of the Literature. *Educational Directions*, December 1985.

7. SQL/DATA System Application Programming for Use, *IBM* 4th edition, December, 1984.

8. Yura, Helen and Walsh, Mary B. *The Nursing Process*. 4th edition. New York: Appleton-Century-Crofts, 1983.

Michael W. Sachon, Systems Analyst
Patricia Damon-Johnson, RDH, MS,
Director of Clinical Affairs
Center for Instructional Development
Computer-Based Learning Laboratory
Old Dominion University
Norfolk, Virginia 23529
(804) 440-3187 or 440-4310



NCS Trans Opt [illegible]-2227 321 A2202

STUDENT NUMBER	COURSE	TODAY'S DATE	DAY	YEAR	APPT NO
		JAN ○			
		FEB ○			
⓪ ⓪ ⓪ ⓪ ⓪ ⓪ ⓪ ⓪ ⓪	0 ○	MAR ○	⓪ ⓪	⓪	⓪ ⓪
① ① ① ① ① ① ① ① ①	1 ○	APR ○	① ①	①	① ①
② ② ② ② ② ② ② ② ②	2 ○	MAY ○	② ②	②	② ②
③ ③ ③ ③ ③ ③ ③ ③ ③	3 ○	JUN ○	③ ③	③	③ ③
④ ④ ④ ④ ④ ④ ④ ④ ④	4 ○	JUL ○	④	④	④
⑤ ⑤ ⑤ ⑤ ⑤ ⑤ ⑤ ⑤ ⑤	5 ○	AUG ○	⑤	⑤	⑤
⑥ ⑥ ⑥ ⑥ ⑥ ⑥ ⑥ ⑥ ⑥	6 ○	SEP ○	⑥	⑥	⑥
⑦ ⑦ ⑦ ⑦ ⑦ ⑦ ⑦ ⑦ ⑦	7 ○	OCT ○	⑦	⑦	⑦
⑧ ⑧ ⑧ ⑧ ⑧ ⑧ ⑧ ⑧ ⑧	8 ○	NOV ○	⑧	⑧ ⑧	⑧
⑨ ⑨ ⑨ ⑨ ⑨ ⑨ ⑨ ⑨ ⑨	9 ○	DEC ○	⑨	⑨ ⑨	⑨

PATIENT NUMBER	PT BIRTHDAY	YEAR	SEX	CLINIC ACTIVITY	INST NO
	JAN ○				
	FEB ○				
⓪ ⓪ ⓪ ⓪ ⓪ ⓪ ⓪ ⓪ ⓪	MAR ○	⓪ ⓪	Ⓜ		⓪ ⓪
① ① ① ① ① ① ① ① ①	APR ○	① ①	Ⓕ		① ①
② ② ② ② ② ② ② ② ②	MAY ○	② ②		CLINIC AST ○	② ②
③ ③ ③ ③ ③ ③ ③ ③ ③	JUN ○	③ ③			③ ③
④ ④ ④ ④ ④ ④ ④ ④ ④	JUL ○	④ ④			④ ④
⑤ ⑤ ⑤ ⑤ ⑤ ⑤ ⑤ ⑤ ⑤	AUG ○	⑤ ⑤		COMP SERV ○	⑤ ⑤
⑥ ⑥ ⑥ ⑥ ⑥ ⑥ ⑥ ⑥ ⑥	SEP ○	⑥ ⑥			⑥ ⑥
⑦ ⑦ ⑦ ⑦ ⑦ ⑦ ⑦ ⑦ ⑦	OCT ○	⑦ ⑦			⑦ ⑦
⑧ ⑧ ⑧ ⑧ ⑧ ⑧ ⑧ ⑧ ⑧	NOV ○	⑧ ⑧			⑧ ⑧
⑨ ⑨ ⑨ ⑨ ⑨ ⑨ ⑨ ⑨ ⑨	DEC ○	⑨ ⑨			⑨ ⑨

PATIENT CLASS				TIME-IN		TIME-OUT		STATUS
CAL	PER	STA	DEN	HR	MIN	HR	MIN	
1 ○	I ○	1 ○	D ○	8 ○	⓪ ⓪	8 ○	⓪ ⓪	
2 ○	II ○	2 ○	M ○	9 ○	① ①	9 ○	① ①	COMP ○
3 ○	III ○	3 ○	P ○	10 ○	② ②	10 ○	② ②	
4 ○	IV ○	4 ○	E ○	11 ○	③ ③	11 ○	③ ③	
5 ○				12 ○	④ ④	12 ○	④ ④	INC ○
				1 ○	⑤ ⑤	1 ○	⑤ ⑤	
				2 ○	⑥	2 ○	⑥	
				3 ○	⑦	3 ○	⑦	
				4 ○	⑧	4 ○	⑧	
				5 ○	⑨	5 ○	⑨	

OLD DOMINION UNIVERSITY

PATIENT NAME:

STUDENT NAME: PRINT LAST NAME FIRST MI

INSTRUCTOR SIGNATURE

DEPARTMENT OF DENTAL HYGIENE AND DENTAL ASSISTING
NORFOLK, VIRGINIA 23508

ASSESSMENT

- Ⓤ Ⓜ Ⓝ MEDICAL/DENTAL HISTORY
- Ⓤ Ⓜ Ⓝ ORAL EXAMINATION
- Ⓤ Ⓜ Ⓝ DENTAL/PERIODONTAL CHARTING
- Ⓤ Ⓜ Ⓝ PERIODONTAL ASSESSMENT
- Ⓤ Ⓜ Ⓝ REASSESSMENT

PLANNING

- Ⓤ Ⓜ Ⓝ TREATMENT PLANNING
- Ⓤ Ⓜ Ⓝ DISEASE CONTROL PLANNING

IMPLEMENTATION

- Ⓤ Ⓜ Ⓝ DISEASE CONTROL INSTRUCTION
- Ⓤ Ⓜ Ⓝ SCALING/ROOT PLANING
- Ⓤ Ⓜ Ⓝ PLAQUE/STAIN REMOVAL

RADIOGRAPHS

- Ⓤ Ⓜ Ⓝ ASSESS OF PATIENT NEED
- Ⓤ Ⓜ Ⓝ INTERPRETATION/UTILIZATION
- Ⓤ Ⓜ Ⓝ TECHNIQUE

FMX ○ EXO ○ OCC ○ PAN ○
PA ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨
BW ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨

COMPREHENSIVE SERVICES

AMALGAM POLISHING	I	① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨
	II	① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨
	V	① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨
SEALANTS		① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨

- ○ CARIES ACTIVITY TEST
- ○ DESENSITIZATION
- ○ FLUORIDE TREATMENT
- ○ IMPRESSIONS
- ○ INTRAORAL PHOTOGRAPHS
- ○ MOUTH GUARD
- ○ NUTRITIONAL COUNSELING
- ○ PEER REVIEW
- ○ PERIODONTAL PACK
- ○ PHASE MICROSCOPE
- ○ PROSTHETIC CLEANING
- ○ PULP VITALITY TEST
- ○ RUBBER DAM APPLICATION
- ○ STUDY MODEL
- ○ TOPICAL ANESTHESIA
- ○ ULTRASONIC SCALING

EVALUATION/MANAGEMENT

- Ⓤ Ⓜ Ⓝ EVAL. OF D.H. TREATMENT
- Ⓤ Ⓜ Ⓝ CLINIC ASSISTANT
- Ⓤ Ⓜ Ⓝ DECISION MAKING
- Ⓤ Ⓜ Ⓝ PREV. DISEASE TRANSMISSION
- Ⓤ Ⓜ Ⓝ TIME MANAGEMENT

TECHNIQUE EVALUATION

GENERAL

- Ⓨ Ⓝ Seeks assistance when needed
- Ⓨ Ⓝ Uses approp. precautions to prevent dis. transmission
- Ⓨ Ⓝ Completes record
- Ⓨ Ⓝ Uses appropriate symbols/terminology
- Ⓨ Ⓝ Explains purpose & findings to patient
- Ⓨ Ⓝ Records all findings & procedures legibly in ink
- Ⓨ Ⓝ Uses proper patient/operator/equipment positions
- Ⓨ Ⓝ Uses adequate direct & indirect lighting

HEALTH HISTORY

- Ⓨ Ⓝ Elaborates on positive responses
- Ⓨ Ⓝ Id. conditions which contraindicate/modify treatment
- Ⓨ Ⓝ Records all required information or medications
- Ⓨ Ⓝ Updates records each appointment

ORAL EXAMINATION/CHARTING

Ⓨ Ⓝ Records deviations/variations	Ⓨ Ⓝ Perio. Assess.
Ⓨ Ⓝ Extraoral	Ⓨ Ⓝ Perio. Charting.
Ⓨ Ⓝ Soft tissue	Ⓨ Ⓝ Dental Charting.
Ⓨ Ⓝ Calculus	Ⓨ Ⓝ Occlusion.

INSTRUMENTATION

- Ⓨ Ⓝ Uses sharp instrument.
- Ⓨ Ⓝ Avoids tissue trauma.
- Ⓨ Ⓝ Makes definitive decisions.
- Ⓨ Ⓝ Uses correct scaling technique.
- Ⓨ Ⓝ Uses correct exploring technique.
- Ⓨ Ⓝ Selects appropriate armamentaria.
- Ⓨ Ⓝ Uses appropriate grasp & fulcrum.
- Ⓨ Ⓝ Establishes logical sequence.

PLAQUE REMOVAL

- Ⓨ Ⓝ Adapts rubber cup & brush to tooth/root surface.
- Ⓨ Ⓝ Uses proper speed and pressure.
- Ⓨ Ⓝ Selects appropriate polishing agent.
- Ⓨ Ⓝ Uses dental floss.

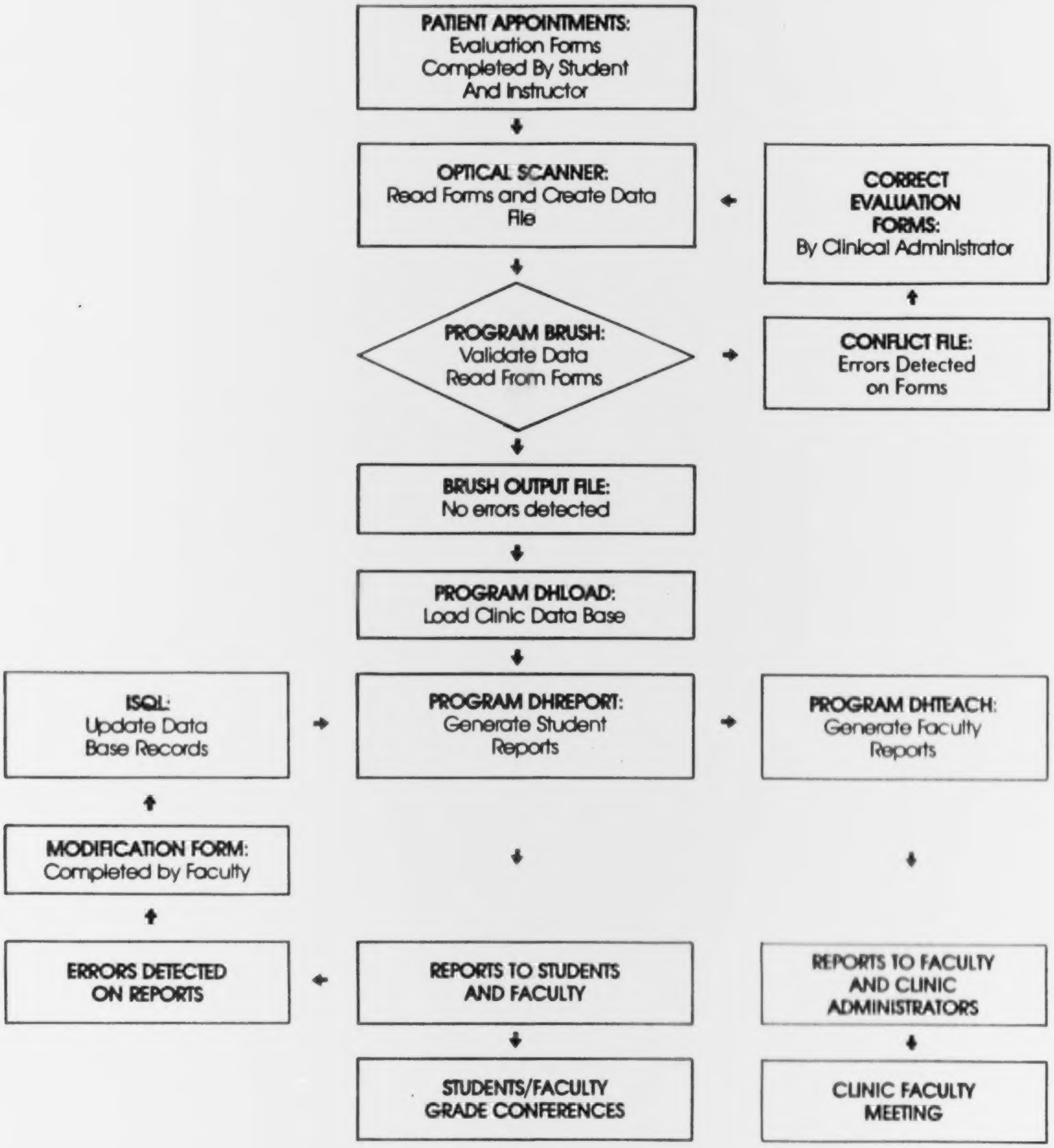
TREATMENT PLANNING

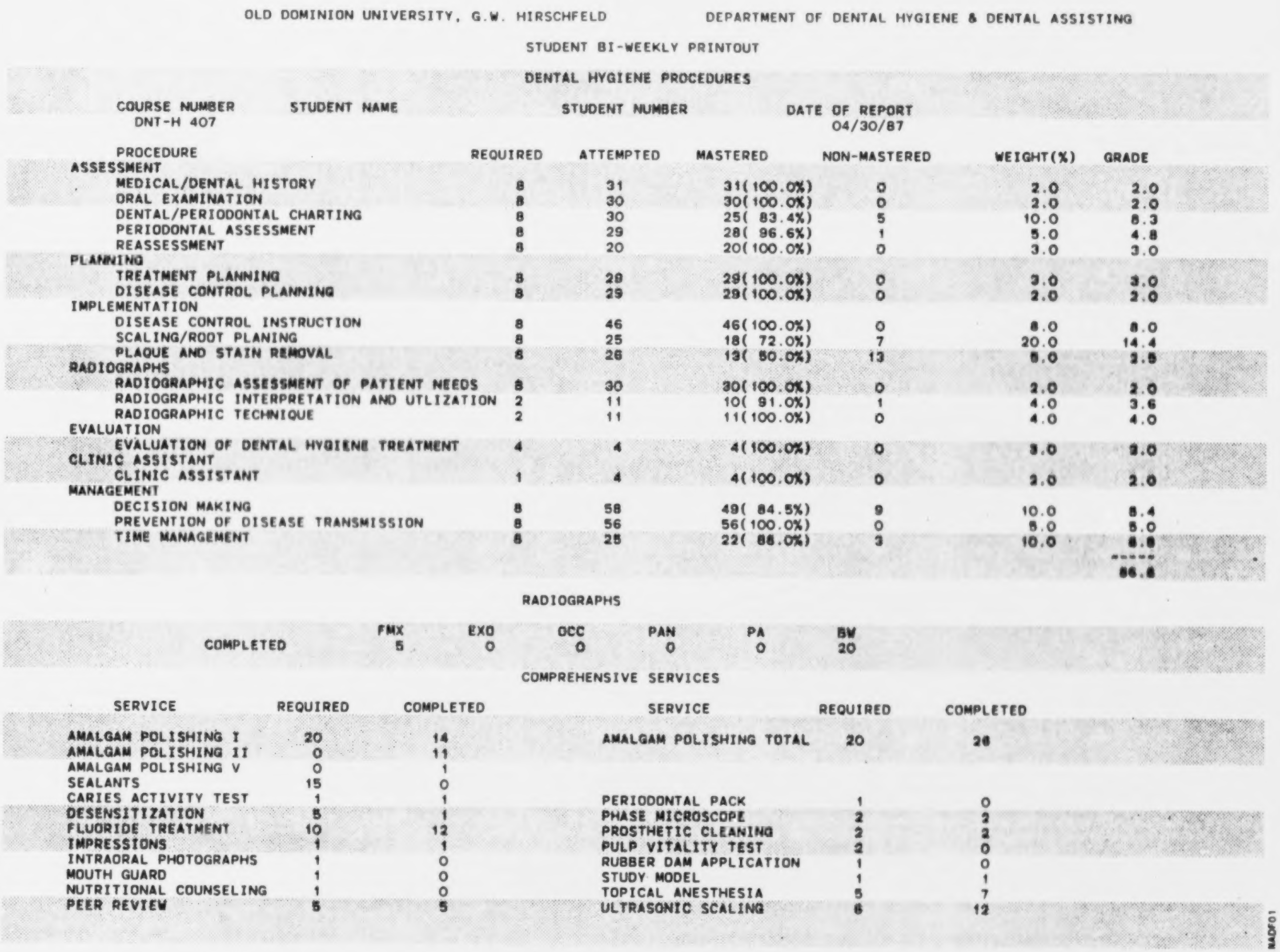
- Ⓨ Ⓝ Establishes & implements comprehensive plan.
- Ⓨ Ⓝ Establishes logical sequence of treatment.
- Ⓨ Ⓝ Discusses plan with instructor & patient.
- Ⓨ Ⓝ Makes alterations in plan when indicated.

SPECIAL CONSIDERATIONS

- Ⓨ Excessive hemorrhage
- Ⓨ Sensitive gagging response
- Ⓨ Tenacious calculus
- Ⓨ Sensitive tooth surface
- Ⓨ Malposed/missing teeth
- Ⓨ Special patient management

FLOW CHART FOR THE COMPUTERIZED CLINICAL EVALUATION SYSTEM

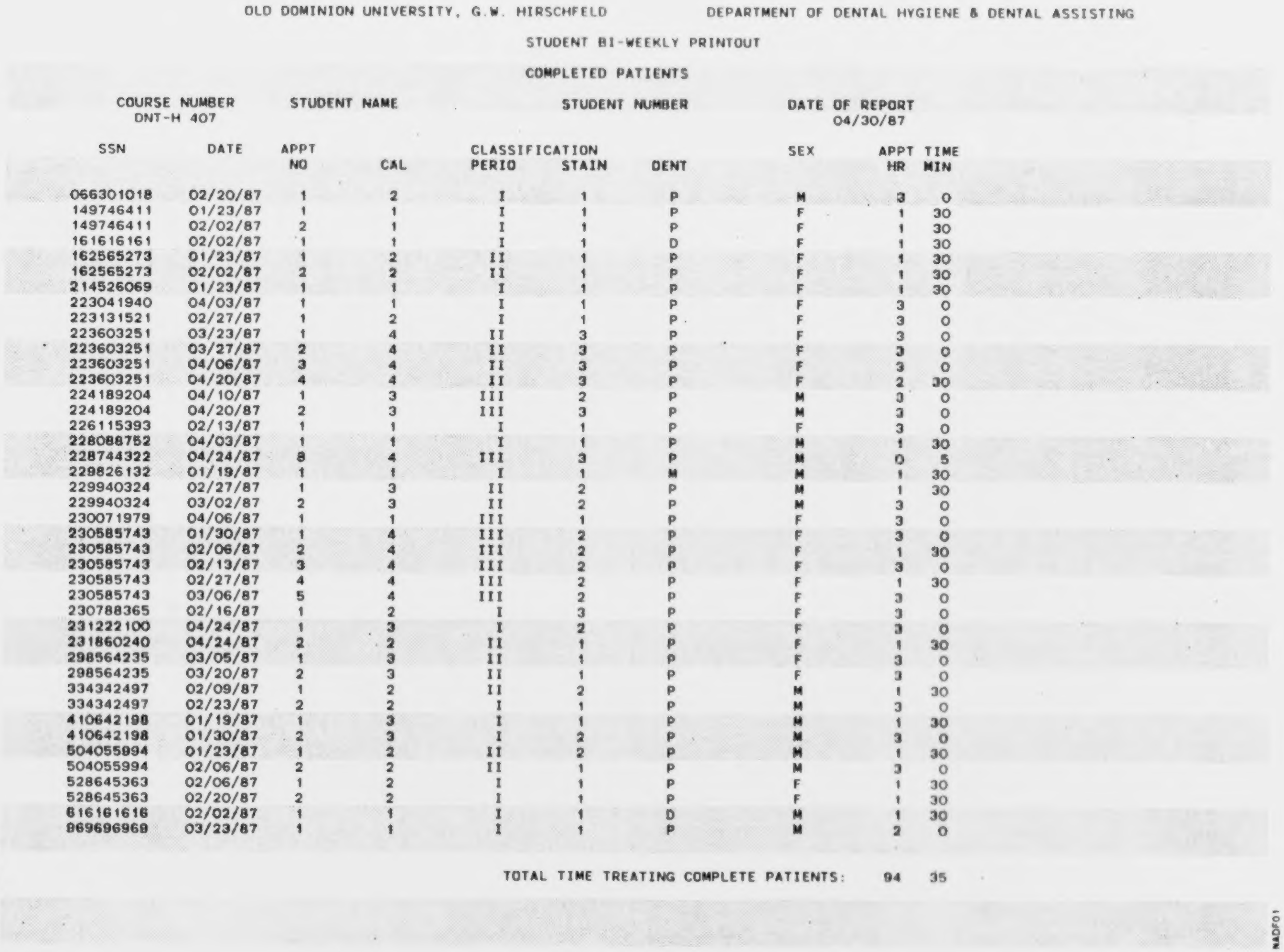

PATIENT APPOINTMENTS: Evaluation Forms Completed By Student And Instructor
OPTICAL SCANNER: Read Forms and Create Data File
CORRECT EVALUATION FORMS: By Clinical Administrator
PROGRAM BRUSH: Validate Data Read From Forms
CONFLICT FILE: Errors Detected on Forms
BRUSH OUTPUT FILE: No errors detected
PROGRAM DHLOAD: Load Clinic Data Base
ISQL: Update Data Base Records
PROGRAM DHREPORT: Generate Student Reports
PROGRAM DHTEACH: Generate Faculty Reports
MODIFICATION FORM: Completed by Faculty
ERRORS DETECTED ON REPORTS
REPORTS TO STUDENTS AND FACULTY
REPORTS TO FACULTY AND CLINIC ADMINISTRATORS
STUDENTS/FACULTY GRADE CONFERENCES
CLINIC FACULTY MEETING



```
OLD DOMINION UNIVERSITY, G.W. HIRSCHFELD          DEPARTMENT OF DENTAL HYGIENE & DENTAL ASSISTING

                                     STUDENT BI-WEEKLY PRINTOUT

                                     DENTAL HYGIENE PROCEDURES

       COURSE NUMBER      STUDENT NAME             STUDENT NUMBER           DATE OF REPORT
        DNT-H 407                                                              04/30/87

       PROCEDURE                                 REQUIRED   ATTEMPTED   MASTERED     NON-MASTERED    WEIGHT(%)  GRADE
ASSESSMENT
       MEDICAL/DENTAL HISTORY                        8         31       31(100.0%)        0             2.0       2.0
       ORAL EXAMINATION                              8         30       30(100.0%)        0             2.0       2.0
       DENTAL/PERIODONTAL CHARTING                   8         30       25( 83.4%)        5            10.0       8.3
       PERIODONTAL ASSESSMENT                        8         29       28( 96.6%)        1             5.0       4.8
       REASSESSMENT                                  8         20       20(100.0%)        0             3.0       3.0
PLANNING
       TREATMENT PLANNING                            8         29       29(100.0%)        0             3.0       3.0
       DISEASE CONTROL PLANNING                      8         29       29(100.0%)        0             2.0       2.0
IMPLEMENTATION
       DISEASE CONTROL INSTRUCTION                   8         46       46(100.0%)        0             8.0       8.0
       SCALING/ROOT PLANING                          8         25       18( 72.0%)        7            20.0      14.4
       PLAQUE AND STAIN REMOVAL                      8         26       13( 50.0%)       13             5.0       2.5
RADIOGRAPHS
       RADIOGRAPHIC ASSESSMENT OF PATIENT NEEDS      8         30       30(100.0%)        0             2.0       2.0
       RADIOGRAPHIC INTERPRETATION AND UTLIZATION    2         11       10( 91.0%)        1             4.0       3.6
       RADIOGRAPHIC TECHNIQUE                        2         11       11(100.0%)        0             4.0       4.0
EVALUATION
       EVALUATION OF DENTAL HYGIENE TREATMENT        4          4        4(100.0%)        0             3.0       3.0
CLINIC ASSISTANT
       CLINIC ASSISTANT                              1          4        4(100.0%)        0             2.0       2.0
MANAGEMENT
       DECISION MAKING                               8         58       49( 84.5%)        9            10.0       8.4
       PREVENTION OF DISEASE TRANSMISSION            8         56       56(100.0%)        0             5.0       5.0
       TIME MANAGEMENT                               8         25       22( 88.0%)        3            10.0       8.8
                                                                                                                -----
                                                                                                                 86.8

                                              RADIOGRAPHS

                                FMX      EXO      OCC      PAN      PA      BW
              COMPLETED          5        0        0        0        0      20

                                         COMPREHENSIVE SERVICES

       SERVICE               REQUIRED     COMPLETED                 SERVICE                REQUIRED     COMPLETED

AMALGAM POLISHING I             20           14          AMALGAM POLISHING TOTAL     20            26
AMALGAM POLISHING II             0           11
AMALGAM POLISHING V              0            1
SEALANTS                        15            0
CARIES ACTIVITY TEST             1            1          PERIODONTAL PACK             1             0
DESENSITIZATION                  5            1          PHASE MICROSCOPE             2             2
FLUORIDE TREATMENT              10           12          PROSTHETIC CLEANING          2             2
IMPRESSIONS                      1            1          PULP VITALITY TEST           2             0
INTRAORAL PHOTOGRAPHS            1            0          RUBBER DAM APPLICATION       1             0
MOUTH GUARD                      1            0          STUDY MODEL                  1             1
NUTRITIONAL COUNSELING           1            0          TOPICAL ANESTHESIA           5             7
PEER REVIEW                      5            5          ULTRASONIC SCALING           6            12
```

Old Dominion University

NADF01



```
OLD DOMINION UNIVERSITY, G.W. HIRSCHFELD          DEPARTMENT OF DENTAL HYGIENE & DENTAL ASSISTING

                                   STUDENT BI-WEEKLY PRINTOUT

                                       COMPLETED PATIENTS

      COURSE NUMBER    STUDENT NAME              STUDENT NUMBER         DATE OF REPORT
       DNT-H 407                                                           04/30/87

      SSN          DATE     APPT              CLASSIFICATION                 SEX      APPT TIME
                             NO      CAL     PERIO     STAIN     DENT                 HR  MIN

   066301018    02/20/87     1        2        I         1         P          M        3    0
   149746411    01/23/87     1        1        I         1         P          F        1   30
   149746411    02/02/87     2        1        I         1         P          F        1   30
   161616161    02/02/87     1        1        I         1         D          F        1   30
   162565273    01/23/87     1        2        II        1         P          F        1   30
   162565273    02/02/87     2        2        II        1         P          F        1   30
   214526069    01/23/87     1        1        I         1         P          F        1   30
   223041940    04/03/87     1        1        I         1         P          F        3    0
   223131521    02/27/87     1        2        I         1         P          F        3    0
   223603251    03/23/87     1        4        II        3         P          F        3    0
   223603251    03/27/87     2        4        II        3         P          F        3    0
   223603251    04/06/87     3        4        II        3         P          F        3    0
   223603251    04/20/87     4        4        II        3         P          F        2   30
   224189204    04/10/87     1        3        III       2         P          M        3    0
   224189204    04/20/87     2        3        III       3         P          M        3    0
   226115393    02/13/87     1        1        I         1         P          F        3    0
   228088752    04/03/87     1        1        I         1         P          M        1   30
   228744322    04/24/87     8        4        III       3         P          M        0    5
   229826132    01/19/87     1        1        I         1         M          M        1   30
   229940324    02/27/87     1        3        II        2         P          M        1   30
   229940324    03/02/87     2        3        II        2         P          M        3    0
   230071979    04/06/87     1        1        III       1         P          F        3    0
   230585743    01/30/87     1        4        III       2         P          F        3    0
   230585743    02/06/87     2        4        III       2         P          F        1   30
   230585743    02/13/87     3        4        III       2         P          F        3    0
   230585743    02/27/87     4        4        III       2         P          F        1   30
   230585743    03/06/87     5        4        III       2         P          F        3    0
   230788365    02/16/87     1        2        I         3         P          F        3    0
   231222100    04/24/87     1        3        I         2         P          F        3    0
   231860240    04/24/87     2        2        II        1         P          F        1   30
   298564235    03/05/87     1        3        II        1         P          F        3    0
   298564235    03/20/87     2        3        II        1         P          F        3    0
   334342497    02/09/87     1        2        II        2         P          M        1   30
   334342497    02/23/87     2        2        I         1         P          M        3    0
   410642198    01/19/87     1        3        I         1         P          M        1   30
   410642198    01/30/87     2        3        I         2         P          M        3    0
   504055994    01/23/87     1        2        II        2         P          M        1   30
   504055994    02/06/87     2        2        II        1         P          M        3    0
   528645363    02/06/87     1        2        I         1         P          F        1   30
   528645363    02/20/87     2        2        I         1         P          F        1   30
   616161616    02/02/87     1        1        I         1         D          M        1   30
   969696969    03/23/87     1        1        I         1         P          M        2    0

                                    TOTAL TIME TREATING COMPLETE PATIENTS:     94   35
```

Old Dominion University

NADF01

DENTAL HYGIENE
FACULTY EVALUATION FREQUENCY

OPSCAN SHEET COUNT = 296
INSTRUCTOR NUMBER = 4
COURSE NUMBER = 5

ASSESSMENT	UNGRADED %	MASTERY %	NON-MASTERY %
Medical/Dental History	69.9%	28.4%	1.7%
Oral Examination	72.0%	25.3%	2.7%
Dental/Periodontal Charting	62.5%	24.3%	13.2%
Periodontal Assessment	55.3%	14.5%	30.2%
Reassessment	67.9%	27.7%	4.4%
PLANNING			
Treatment Planning	68.2%	30.4%	1.4%
Disease Control Planning	38.2%	57.1%	4.7%
IMPLEMENTATION			
Disease Control Instruction	22.7%	66.9%	10.4%
Scaling/Root Planing	55.4%	24.4%	20.2%
Plaque/Stain Removal	55.4%	42.2%	2.4%
EVALUATION			
Evaluation of Dental Hygiene Treatment	94.6%	5.1%	0.3%
MANAGEMENT			
Clinic Assistant	0.0%	0.0%	0.0%
Decision Making	31.1%	53.9%	15.0%
Prevention of Disease Transmission	42.0%	54.8%	3.2%
Time Management	67.9%	28.0%	4.1%
RADIOGRAPHS			
Assessment of Patient Need	72.0%	26.2%	1.8%
Interpretation & Utilization	76.7%	22.0%	1.4%
Technique	76.7%	15.0%	8.3%

INNOVATIVE USE OF PLATO NOTEFILE IN DIETETIC CLINICAL SIMULATION

Leta P. Aljadir, M.S., R.D.
University of Delaware

Evelyn V. Stevens, Ph.D.
University of Delaware

ABSTRACT

The PLATO Notefile at the University of Delaware was successfully adapted for developing an inexpensive computerized clinical case simulation for dietetics students. This teaching tool provides a quasi-clinical experience through which students develop decision-making skills. A discussion session at the completion of the case permits analysis of case management decisions and case outcomes.

INTRODUCTION

The use of the computer in clinical education originated in the late 1950's and early 1960's, and was first applied in the medical field (Engle, 1963; Ledley and Lusted, 1959; Warner, et al 1961). Initial computer applications in the medical sciences consisted primarily of data management, as in clinical laboratory data and patient monitoring. The recognition that medical diagnosis is subject to mathematical analysis, hence information processing by computer, widened the application of computers in medicine. Its outgrowth into clinical education soon followed.

Much of medical and allied health education deals with skill-building, in addition to the traditional acquisition of a sound knowledge base. The skilled clinician is adept at decision-making, using various sets of data bases as a case proceeds, to ensure the best possible outcome. The acquisition and refinement of skills is accomplished through labor-intensive clinical instruction. This phase of clinical education usually reflects an extremely low student to clinical instructor ratio and is, therefore, very expensive.

Computerized clinical simulations are an attractive adjunct to clinical instruction. There are a number of medical and allied health schools which have adopted clinical simulations for medical diagnosis and case management (Barnett, 1968; Barnett, Hoffer and Famiglietti, 1968; Harless, Zier and Duncan 1986; Friedman, 1986; Friedman et al, 1978; Jones and Keith, 1983; Wigton, 1986). Although most experts agree that these computerized clinical simulations incur phenomenal developmental costs, several advantages offset this initial investment. For instance, clinical simulations allow the student to manipulate theoretical cases without risk to the patient; the collection of simulations may include unusual cases to which the student may then be exposed; the clinical "experience" of the student is not limited to the cases available at the clinical site at any given time; simulations provide an opportunity to instructors to standardize evaluation of the students' clinical skills (Jones and Keith, 1983).

Clinical education in dietetics, much like any other clinical education, is time and labor intensive. The adaptation of the medical model of computerized clinical simulations to dietetics holds much promise of effective use of computer-based education (Breese, et al, 1977; Schroeder and Kent, 1982). However, the high cost of elaborate clinical simulations precludes their use in small departments.

An alternative model of clinical simulation was developed at the Department of Nutrition and Dietetics, and the Office of Instructional Technology at the University of Delaware utilizing an existing computerized program (i.e. PLATO notefile) to provide a quasi-clinical experience for dietetics students.

The Department of Nutrition and Dietetics offers a Coordinated Undergraduate Program in Dietetics (CUPD) in which clinical experiences are integrated into the curriculum at the junior and senior years. A traditional dietetics curriculum is also offered, which does not include a clinical program. The clinical simulation using the PLATO notefile was introduced primarily for the benefit of students in the latter program.

The PLATO student notefile is part of a larger electronic communication and mail system. This particular part of the PLATO system allows the student and instructor to communicate with each other. Students do not have access to each other's notes and responses. Their work on the computer is, thus, individualized and confidential.

The clinical simulation consists of an initial case scenario which elicits certain responses from each student. The instructor then adjusts the case as each student's response dictates. The process goes through a predetermined number of iterations to arrive at the final outcome of the case. A hard copy is made of the initial case and all the students' responses, and is distributed to all the students at a final discussion session. At this time, the various twists to the case resolutions are made evident to the students, highlighting the divergent paths taken based on the students' individual decision-making. This allows some discussion of decision analysis, which is crucial to clinical judgement and case management.

SYNOPSIS OF SAMPLE CLINICAL SIMULATION

This clinical simulation was used during two different semesters and offered to a limited number of students (n = 10 per semester) on a first come, first served basis. Students were given approximately 1-1/2 weeks to react and respond to the changing scenarios. The instructor was also allotted 1-1/2 weeks to read each response and adjust the case scenario based on student responses. A brief description of the simulation follows.

Initial Case Scenario: Description of a college student experiencing weight problems. Students are asked to request further information and estimate daily calorie intake to accomplish a safe weight loss pattern.

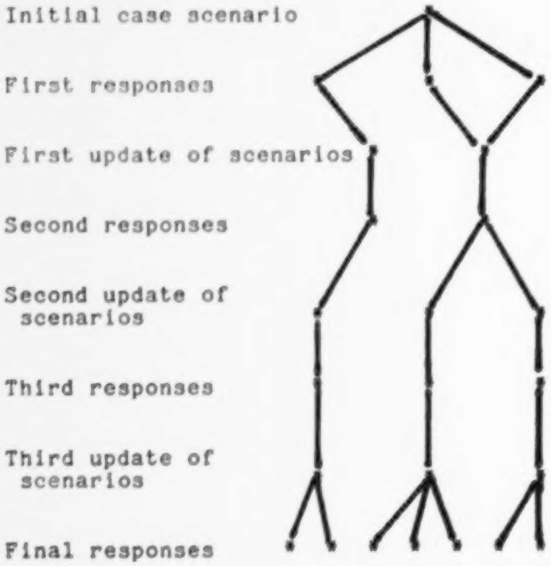
(First round of student responses.)
First Update on Case Scenario: Information provided as per student request; students asked to prescribe diet by exchange lists and to provide initial counseling.

(Second round of student responses.)
Second Update on Case Scenario: Description of progress or lack thereof at 4 weeks. Students asked to help client be more successful with weight loss program.

(Third round of student responses.)
Third Update of Case Scenario: Description of progress or lack thereof at 10 weeks. Students asked to evaluate success or failure of weight reduction program and prepare for final group discussion.

There were several different weight loss patterns predicted by the students based on the information presented to them. The branching pattern for this simulation is shown in Figure 1.

Figure 1. BRANCHING PATTERN IN DIETETICS CASE SIMULATION



One of the primary learning outcomes in this exercise was the demonstration of the different paths and case resolutions possible with a single case, depending on decision making. In the final discussion group, students were able to see how each of their decisions impacted the outcome of the case.

The students were asked to evaluate the clinical simulations using the PLATO notefile as a quasi-clinical experience. Four of the ten students who participated in this activity during the fall of 1986 were students in the CUPD (i.e. clinical program); six of the students were in the non-clinical dietetics program. Without exception, the students believed that the exercise was valuable and gave them some exposure to decision-making and case management. The CUPD students, however, believed that since the number of students participating in the activity had to be limited, the non-clinical dietetics students should have been given preference. This was in view of the fact that the CUPD students are exposed to the clinical setting in their curriculum.

The overall evaluation of the activity can be summarized in terms of advantages and disadvantages to the student and to the instructor. This is presented in Table 1. The major advantage for students was the provision of an individualized, self-paced simulated clinical experience. In contrast to the elaborate computerized clinical simulations, the PLATO notefile provides a means for the instructor to quickly develop as simple or elaborate a case as dictated by instructional demands.

In conclusion, it appears that the PLATO notefile is a viable mode for developing low cost clinical simulations. Informal evaluation of the activity by student users indicates a positive attitude toward this teaching tool.

Table 1: ADVANTAGES AND DISADVANTAGES OF CLINICAL SIMULATIONS UTILIZING THE PLATO NOTEFILE

	Advantages	Disadvantages
Instructor	Time economy Enables instructor to monitor student's application of theory without use of the clinical setting and without scheduled conferences Low cost Utilizes existing PLATO system Provides a growing source of new material to share with future generations of students; cases may be as simple or as elaborate as instructor decides	Time spent reading responses and updating scenarios Limited applications (e.g. this format does not lend itself to clinical cases requiring short turn-around time)
Students	Simulates clinical cases (e.g. CUPD students attest to simulation of clinical management of cases) Allows student to apply theoretical knowledge; experimentation with various decision paths is possible Time economy Individualized pace and schedule	Not a real clinical experience Occasional delay in response time by instructor, with attendant loss of interest by student Limited application

REFERENCES

1. Barnett, G.O. (1968). Computers in patient care. New England Journal of Medicine, 279:1321-1327.

2. Barnett, G.O., Hoffer, E.P., Famiglietti, K.T. (1986). Computer-based patient simulations. In H.F. Orthner (Ed.), Proceedings. The Tenth Annual Symposium on Computer Applications in Medical Care (pp. 176-179). Computer Society Press.

3. Breese, M.S., Welch, A.C., and Schimpfhauser, F. (1977). Computer-simulated clinical encounters. Journal of the American Dietetic Association, 70:382-388.

4. Engle, R.L., Jr. (1963). Medical diagnosis: present, past, and future. III. Diagnosis in future, including critique on use of electronic computers as diagnostic aids to physician. Archives of Internal Medicine 112:530-543.

5. Friedman, R.B. (1986). Computer based patient simulations for testing clinical competency. In H.R. Orthner (Ed.), Proceedings, The Tenth Annual Symposium on Computer Applications in Medical Care (pp. 188-190). Computer Society Press.

6. Friedman, R.B., Korst, D.R., Schultz, J.V., Beatty, E., Entine, S.M. (1978). Experience with the simulated patient-physician encounter. Journal of Medical Education, 53:825-830.

7. Harless, W.G., Zier, M.A., Duncan, R.C. (1986). Interactive videodisc case studies for medical education. In H.F. Orthner (Ed.), Proceedings. The Tenth Annual Symposium on Computer Applications in Medical Care (pp. 183-187). Computer Society Press.

8. Jones, G.L., and Keith, K.D. (1983). Computer clinical simulations in health sciences. Journal of Computer-Based Instruction, 9(3):108-114.

9. Ledley, R.S., and Lusted, L.B. (1959). Reasoning foundations of medical diagnosis: symbolic logic, probability, and value theory aid our understanding of how physicians reason. Science, 130:9-21.

10. Schroeder, L., and Kent, P. (1982). Computer-based instruction in dietetic education. Journal of Computer-Based Instruction, 8(4):85-90.

11. Warner, H.R., Toronto, A.F., Veasy, L.G., and Stephenson, R. (1961). Mathematical approach to medical diagnosis: application to congenital heart disease. Journal of the American Medical Association, 177:177-183.

12. Wigton, R.S. (1986). Use of computer simulations in teaching clinical diagnosis. In H.F. Orthner (Ed.), Proceedings. The Tenth Annual Symposium on Computer Applications in Medical Care (pp. 180-182).

Leta P. Aljadir
Department of Nutrition and Dietetics
University of Delaware
Newark, DE 19716
302-451-2139/451-8976

IMPLEMENTATION OF A THREE DIMENSIONAL ANIMATION MODULE IN DESIGNERS SPATIAL

Susan Harshbarger, Illinois State U.

ABSTRACT

Educators are engaged in the training of architects and designers in the fundamentals and advanced concepts of Computer-Aided drafting/design (CADD). Spatial skills are essential skills for interpreting multidimensional concepts as well as contributing to design creativity. To assist in the development of spatial skills Illinois State University introduced in 1986 a computer generated three dimensional graphics animation module into the interior design program. Instructional materials and skills were developed to aid in the educational process of utilizing the CADD modules.

INTRODUCTION

An objective of educational institutions is to stay on the leading edge of computer technology. Educators are engaged in the training of architects and designers in the fundamentals and advanced concepts of Computer-Aided drafting/design (CADD). The ability of architects and designers to communicate graphically on a CADD has been necessary to keep abreast of technological innovations in the design field (Hoffman, 1983; Hernecheck, Rettig and Sherman, 1983; and Myers, 1982). By exposing students to a CADD system they are obtaining hands-on experiences with state of the art automated systems. Since a CADD system with its massive memory and permanent storage allows a person to plot out a design with tremendous speed one can examine various design solutions that would not have been possible in the past because of time and conceptual constraints. Research with interior design students by Sherman and Hill (1984) indicated that the absence of hands-on experience with the CADD system may be disadvantageous to the students. Furthermore the study determined that it is important for an individual to know how to apply the automated system and not merely to learn what the system could do.

Spatial skills are essential skills for drafting as well as contributing to design creativity (Karlans, Schuenhoff and Kaplan, 1969). Visuo-Spatial ability is necessary to understand and interpret geometric and multi-dimensional concepts (Lowery and Knirk, 1982; Baker and Belland, 1986). These abilities interact with an understanding of pictorial information on computer screen displays and with one's manipulation of displayed images. Researchers have determined that spatial skills can be taught with hands-on experience (Baker and Belland, 1986; Bishop 1973; Sherman and Hills, 1984; Zavotka, 1986). Timing and the type and structure of activities are important elements of an effective approach to developing spatial skills. CADD systems on high resolution computer generated three dimensional graphics animation offer new devices for developing spatial skills. For the formation of an accurate spatial representation, Evans (1980) found that multiple viewpoints of an object are important. Computer generated graphic animation is helpful in developing structural descriptions that includes the orthographic features of an architectural object by projecting unlimited perspective movements through a mathematically defined three dimensional space. Olson and Bialgstok (1983) reported significant results with direct block manipulation training and visually mediated experiences in lock rotations.

Integration of computer generated three dimensional graphics animation and the interior design program at Illinois State University occurred in the Fall,1986 semester. During this time hands-on use of the Palette three dimensional modeling and Color-Shaded Imaging CADD system provided the professors with the knowledge necessary to teach the computer generated three dimensional graphics animation course materials. In addition instructional materials and skills were developed to aid in the educational process of utilizing the CADD modules.

PROGRAM AND TECHNIQUE DEVELOPMENT

USE OF THE CADD SYSTEM

Three dimensional computer generated graphics animation has several attributes for the educational process of spatial skills. The three dimensional module is designed to build on the two dimensional system. The Palette base two dimensional module is designed to emulate the natural drafting and design process through a use command structure. All commands where practical are nemonic (e.g., M for Move, D for draw, A for advance into three dimensional module, etc.). Commands are imput to the system as groups of keystrokes where the most commonly used commands require the least keystrokes.

The three dimensional module is primarily a design tool allowing the investigation and verification of 3-D spatial relationships. Visuals can be produced in either wireframe or fully three demensional (not $2\frac{1}{2}$D) models. The three dimensional module is also a tool for producing drawings of three dimensional scenes. Several orthographic oblique and isometric projections can be created from a single 3-D model (Figure 1). An interchange of information with the drafting module occurs through rotation, sectioning and walk-throughs. The animation program permits one to visualize objects rotated to various views and to see the change from three dimensions to two.

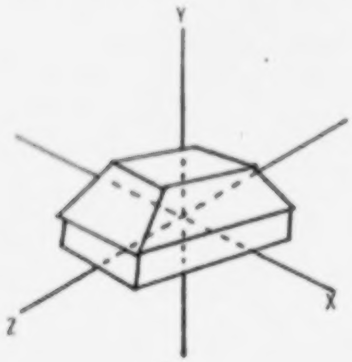
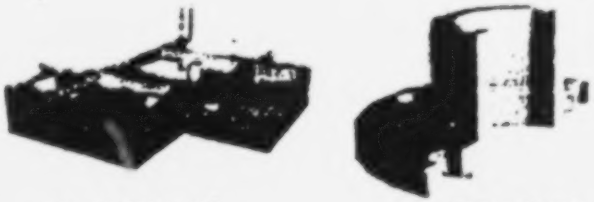


Figure 1 A mansard roof on a house

Orthographic projection with X, Y, and Z axis location indicated.

Color-Shaded Imaging is designed to interact with 3_d models. Models can change from wireframe to shaded images. Models can move up, down, and around with the animation mode. Multiple light sources, transparencies (windows you can see through), fog simulations, and Z-clipping are drafting techniques available in color-shaded imaging (Figure 2).


Interior Office Plan

Exterior Facade

Figure 2
Color-Shaded Imaging Models.

INTEGRATION OF CADD IN THE DESIGN PROCESS

CADD is automation that enhances and expands manual methods and is a tool for transposing visual concepts to computer applications. Three dimensional computer generated graphics animation can produce more detailed illustrations of design models and solids. These attributes contribute toward efficiency in productivity and design implementation by reducing redundancy and repetition in design and drafting.

Efficiency in learning basic spatial skills is crucial to design students. Three dimensional computer generated graphics animation aids the student in acquiring basic spatial skills by creating an efficient framework for mental imaging of orthographic projections. Furthermore the module assists designers in acquiring the spatial skills needed for them to interpret various forms of technical drawings such as cross-sections, plans, elevations, and perspectives. Three dimensional computer generated graphics animation also enables a designer to orient oneself in three-dimensional space at different angles and at different times. The acquisition of these specific spatial skills is an important development of a designer's spatial aptitude.

Critical to the development of a designer's spatial aptitude and skills is appropriate instructional strategies and objectives as well as the necessity for developing course content systematically and carefully. To integrate the three dimensional computer generated graphics animation with interior design course content systematically and carefully. To integrate the three dimensional computer generated graphics animation with interior design course contents (lecture and instructional hand-out materials) substantive print support materials were developed. A faculty resource guide for each design project contains objectives, background information, design criteria, and guidelines for evaluating student solutions. The guide is fully illustrated with the appropriate instructions, drawings specifications, dimensions, and design concepts necessary for the completion of the design projects. Clear instructions to students on how to operate the computer and interact with the three dimensional computer generated graphics animation module are included in the course materials. All materials are highly interactive requiring sophisticated student design solutions and hands-on experiences in order to move sequentially through the various components to complete the projects. Therefore, course content seeks to reinforce the understanding of the design process and principles through experiential learning.

IMPLICATIONS FOR RESEARCH

Researchers need to investigate further how learning spatial skills can be enhanced through educational experiences. Since learning is influenced as much by instructional strategy as it is by mental skills; given that other variables such as knowledge of tasks, form in which information is presented, and general ability all influence learning (Winn, 1982) research in acquiring spatial skills needs to address the questions of cognitive processes and instructional design theories simultaneously. Future studies must also focus on ways to relate the active/reflective cognitive dimension more directly to CADD characteristics in order to incorporate intelligent components for appropriate interactive simulation.

CONCLUSIONS

The potential of CADD with a three dimensional computer generated animation module and other technologies for effective design process interaction has been recognized by educators and intensively studied by industry. It is the intent of this educator to analyze systematic applications of ways in which the state of the art technology can be used to support skills and creativity components of design educational research to obtain strides in the effectiveness and efficiency of the interior design process.

Susan Harshbarger, ISU, Normal, IL 61761
309/438-2517

REFERENCES

Baker, P. and Belland, J. (1986). Using computer displayed pictures to enhance visuo-spatial aptitude in pre-adolescent females: suggestions and opportunities for research. 28th ADCIS Conference Proceedings, Washington, D.C. -- Crystal City, November 10-13, 1986.

Bishop, A.J. (1973). The use of structural apparatus and spatial ability - a possible relationship. Research in Education, 9, 43-49.

Evans, G. and Pezdek, K. (1980). Cognitive mapping: Knowledge of real-world distance and location information. Journal of Experimental Psychology: Human Learning and Memory, 5, 13-24.

Hernecheck, P.J., Rettig, K.D. and Sherman M.P. (1983). Professional viewpoints of competencies for interior design entry-level positions. Journal of Interior Design Education and Research, 9(92),7-13.

Hoffman, C.L. (1983). Architectural graphics competencies in interior design: A comparison of professional and student usage. Journal of Interior Design Education and Research. 9 (1), 14-19.

Karlans, M., Schuerhoff, C. and Kaplan, M. (1969). Some factors related to architectural creativity in graduating architectural students. Journal of Genetic Psychology, 81, 203-215.

Lowery, B. R. and Knirk, F.G. (1982). Micro-computer video games and spatial visualization acquisition. Journal of Educational Technology Systems, 11 (2), 155-166.

Myers, C. (1982). Entry level competencies needed by interior designers. Journal of Interior Design Education and Research, 8(1), 19-24.

Olson, D.R. and Bailystok, E. (1983). Spatial Cognition, Hillsdale, New Jersey: Lawrence Erlbaum Association.

Sherman, M.P. and Hill, S.L. (1984). Computer aided design: from manual to plotter. Journal of Interior Design Education and Research, 10(2), 13-17.

Winn, W. (1982). Visualization in learning and instruction: A cognitive approach. ECTV, 30 (1), 3-25.

Zavotka, S.L. (1986). Analysis of three dimensional computer graphics animation to teach spatial skills to interior design students. Journal of Interior Design Education and Research. 12(1), 45-52.

Susan Harshbarger
Illinois State University
Turner Hall-Home Economics
Normal, IL 61761
(309) 438-2517

UTILIZING AN INTERACTIVE COMPUTER-BASED APPROACH FOR ENHANCING THE STUDY OF FAMILY FINANCIAL PLANNING

Dr. James L. Morrison

University of Delaware

ABSTRACT

This paper focuses on the development of computer software as part of an interactive instructional approach which results in adapting current survey-type personal finance courses into an application's offering. Traditionally, introductory personal finance courses have focused upon presenting to students via survey methodology a variety of money management topics. The purpose of such a personal finance course is to permit students to gain a general knowledge about many topics with perhaps the opportunity of performing indepth study in only one particular aspect of money dynamics. The computer-based educational strategy presented in this paper implies that a modification in the basic philosophy of personal finance courses may be appropriate for meeting the changing needs of business and students.

NEED FOR COMPUTING SOFTWARE

Family financial planning is one of the new service-oriented career options for majors in consumer economics. Although a variety of software has already been developed for assisting families in budgeting, investing, tax preparation, and estate planning, etc., there appears to be little systematic software for instructing students in critical aspects of the financial planning profession itself.

PILOT STUDY COMPLETED

During the past several years, beginning in 1983, a computer-based case study application's module in family financial counseling was initiated, designed, and implemented in a Consumer Financial Management course in the College of Human resources at the University of Delaware. The case study acts as the culminating experience for systhesizing 10 segregated parts of coursework into one dynamic application. A pilot study was conducted during Fall 1986 to determine the impact such a case study had on student learning.

STUDY SAMPLE

Students enrolling in Consumer Financial Management participated in the field testing of a computer module emphasizing an interactive instructional strategy. The undergraduate class met three days a week for 13 weeks from September 1986 to December 1986. Being a large-group section, 39 students (primarily juniors and seniors) completed the computer-based module. Of the 39 students, 30 were female and 9 were male. In addition, 21 were consumer economics majors, 8 were majors in other fields in the College of Human Resources, and 10 were students selecting the course to fulfill a requirement in majors outside of the College of Human Resources.

OBJECTIVES OF THE COMPUTER-BASED MODULE

The application's computer module requires each student to assume the role of a family financial planner. The objective of the computer module is to have the student apply basic money management principles reflecting a variety of money management responsibilities, including a net worth analysis, tax planning, budgeting, recordkeeping, saving, investing, and estate planning. Students are requested to develop a financial plan which adheres to specific criteria reflecting guidelines discussed and analyzed during coursework throughout the semester. By the conclusion of the computer-based module, students are able:

(1) to apply 10 basic money management principles as part of a family financial counseling strategy;

(2) to apply interviewing techniques while composing appropriate questions for getting required information from clients;

(3) to devise a financial plan, justifying its

creation, as measured by agreement to 10 money management principles;

(4) to maintain appropriate records and files for each client.

OVERALL DESIGN OF SOFTWARE

A "menu format" is used in the computer-based module. Inserted below is a "lesson frame" which appears at the terminal as a display. As illustrated, each student selects in proper sequence 10 money management principles. The listing below contains 15 principles in unarranged order. The challenge for students is two-fold: (1) Initially, to place each money management principle in an appropriate order and (2) secondly, to incorporate that money management principle into a financial plan.

LESSON MENU

Listed below are 15 possible steps for the development of a financial strategy. ONLY 10 OF THESE STEPS relate to the 10 financial principles. These steps must be done in their proper order. Your job is to choose the next appropriate step.

a. Complete a personal information sheet
b. Examine investment portfolio
c. Calculate the client's net worth
d. Complete a time analysis
e. Develop a budget
f. Complete an income tax form
g. Select members of an advisory team
h. Establish a records management system
i. Write will and letter of instructions
j. Utilize the compounding interest formula
k. Analyze insurance coverage
l. Perform time value calculations
m. Verify estate planning procedures/ documents
n. Determine priorities
o. Analyze client's income and tax liabilities

Type the letter of the FIRST step.

PROCEDURES OF STUDY

The process followed by each student requires the interviewing of a potential client. For example, students by proposing "free lance" questions must ascertain at the beginning of the case study the needs of the client. Once the data file is developed, an appropriate strategy is designed. The flexibility of the computer module enables students to arrive at different strategies. The financial plans of the students become the basis for the final examination for the course. The emphasis here is upon determining the philosophical, sociological, and economical impacts upon family welfare.

After 12 weeks of coursework and upon completion of activities reflecting the 10 money management principles illustrated in the previous menu, each student was provided a PLATO[1] terminal to complete the assignment. The students were presented with a deadline at which time their final plan was to be submitted to the client. Merit points were accumulated by each student as suggestions made for applying 10 money management principles "fell" within the guidelines. A total of 100 merit points could be achieved. In addition, students had the opportunity to take a posttest which included 10 multiple-choice questions. These questions required the student to analyze the complexities of family financial planning as demonstrated in the module completed. An additional 10 points could be added to their scores, resulting in an overall performance rating.

The capabilities of the PLATO timesharing network permit students to record answers for consideration which may not be perceived as appropriate by the computer program. For example, during the case study, checkpoints were included which asked students specific questions about their strategy. If the student responded within accepted guidelines which were programmed into the lesson, then the student achieved a merit point. However, there may be occasions whereby the student's response was not accepted as correct but may have been. This opportunity to record their "free lance" responses and receive credit added an important dimension to the case study. Using the electronic mail network associated with the PLATO timesharing system, student comments could be responded to quickly, thus providing immediate feedback.

STUDY FINDINGS

In evaluating student performance on the case study, any student who achieved 80 points or higher prior to completing the post-test option was identified as being successful. Any student who received 79 points or less was identified as being unsuccessful. In addition, students overall grade point average (GPA) was matched to performance as measured by the total merit points accumulated by the conclusion of the case study.

Forty-two students participated in the study. The findings indicated that 35 of the 39 students were able to accumulate over 80 merit points by the conclusion of the case study. However, 3 students withdrew from the project and the course. Indicated in Table 1 are frequency counts by 5-point categories.

---

[1] PLATO is a registered trademark of Central Data Corporation

TABLE 1

FREQUENCY COUNTS OF MERIT POINTS ACCUMULATED PER STUDENT

N=39

# of Merit Points	# of students
95 - 100	10
90 - 94	10
85 - 89	8
80 - 84	7
75 - 79	4

In addition, the time constraints of the case study were critical to student planning. Each student was allocated 6 hours of computer access time to complete the activities. Indicated in Table 2 are the calculated completion times of students:

TABLE 2

LENGTH OF TIME TO COMPLETE MODULE

N=39

Minutes	# of students
120 - 180	3
181 - 240	11
241 - 300	22
301 - 360	3

For determining the relationship between the students grade-point average and degree of success when completing the computer-based interactive module on financial planning, the Pearson r and t-test statistical measures were utilized. Using a Pearson Correlation statistical analysis, findings indicate that there is only a very slight positive relationship between the two variables: individual grade-point average and test scores of students on the computer-based module.

During the study, students were grouped into two categories: those with a grade-point average of 2.5 or above (n=20) and those with a grade-point average of 2.4 or below (n=19). When matching mean test scores of each group of students to grade-point average using a t-test statistical analysis, findings generally support the position that there appears to be a great similarity in degree of success of students. The hypothesis was accepted that there was no significant difference between the mean test scores of students by GPA Group (t-score = -.1917; df 37; level of significance .8490**).

Summary performance statistics for each student group are indicated below:

	GPA GROUP 1 (GPA $<$ 2.5 -)	GPA GROUP 2 (GPA $>$ 2.5 +)
	Test Scores	
Mean	88.6	89.0
Median	91.0	89.0
Range - High	99.0	99.0
Low	76.0	75.0
Range Spread	23.0	24.0
$Q_3 - Q_1$	13.0	10.3

During the evaluation process, students were provided the opportunity to write comments. Generally, the two most frequent positive comments related to (1) understanding how all the parts of coursework interrelated and (2) receiving immediate feedback as to rate of success (merit points). Negative comments were fewer in number. The primary negative comments focused upon (1) the pressure of meeting a deadline and (2) the pressure of recording an answer knowing that the individual would immediately be penalized for incorrect response.

CONCLUDING COMMENTS

When matching grade-point average (GPA) of each student to degree of success, the findings indicated that students with GPA's below 2.5 and those 2.5 or above had similar success. Using a t-test for statistical significance, no suggestive difference between the degree of success of students in these two groups was determined based upon GPA.

It appears that the case study in its present format assists students with relatively high and relatively low grade point averages. The interactive design of the case study encourages active student participation which involves more than responding to multiple choice questions. The ability to "free lance" responses to programmed questions encourages the students to think carefully about their advice. The repercussions of giving inappropriate advice comes quickly and decisively.

The replacement of a survey-type personal finance course with an integrated design incorporating family financial counseling appears to be a viable alternative for updating curriculum. The "piecemeal fashion" of traditional personal finance coursework enables students to get only a "flavoring of a variety of topics" with little opportunity to apply learnings to real life situations. This case study module delivered through a timesharing system enables students to apply synthesizing and evaluating skills during the learning process. The success indicated in the findings of the pilot test supports the alternative of integrating money management concepts by utilizing the application's approach to learning.

**not significant at .05 level

ADDRESSING THE COMPUTER LITERACY NEEDS OF HOME ECONOMICS EDUCATORS IN HAWAII

Dr. Cecelia Thompson

University of Arkansas

ABSTRACT

The objectives of this study were to assess computer use by home economics teachers in Hawaii, to develop a computer literacy workbook for home economics educators, and to develop computer literacy through inservice activities.

A survey was sent to all home economics educators in Hawaii to determine their use of the computer. A majority of respondents had access to a computer and some background knowledge of operation and use.

A computer literacy workbook, Introduction to Computers for Home Economics Educators, was developed. The workbook addressed the components and peripherals of a computer, common computer terminology, operating a computer, simple programming, and software selection and use. The workbook was developed on a MacIntosh computer and a laser printer to illustrate desktop publishing.

Six computer workshops emphasized hands-on computer experiences and developing positive attitudes about the computer. The workbook was field tested at each workshop and revised using suggestions from educators. Evaluation of the workshops determined that computer anxiety decreased and knowledge of computer use increased.

This study revealed ways that home economics educators are enhancing the quality of teaching and learning with the use of computers.

ADDRESSING THE COMPUTER LITERACY NEEDS OF HOME ECONOMICS EDUCATORS IN HAWAII

In Hawaii, every school has computers for student and faculty use. Computer awareness classes are integrated into junior high math program and secondary business and math. Little was known about how Hawaii's home economics teachers were using computers. To gather this information, a home economics educator computer questionnaire was mailed to the 124 public school home economics teachers in Hawaii. The questionnaire requested information about access to microcomputers, software and microcomputer use, and interest in learning more about using a microcomputer.

Ninety-three teachers returned the questionnaire. Sixty-two percent of the respondents had access to a microcomputer. Every school had a computer on campus; however, 38% of the teachers indicated the computer was not available for their use.

Seventy-three percent of the respondents had used microcomputers. Many home economics educators had participated in computer awareness programs offered by the Hawaii Department of Education and computer classes at the University of Hawaii.

Most of the respondents were using Apple computers. Fewer were using IBM, followed by Commodore, TRS 80, Atari, and Compaq. The computer was most frequently used for word processing. Grade keeping was the second most popular use. An equal number of respondents indicated using student tutorials and games. Only 3 teachers used the computer for classroom inventory.

When the educators were asked to list by name some of the software they found most useful, word processing programs topped the list. Appleworks and Bankstreet were most frequently listed. Games and graphics programs were also popular. Print Shop was used frequently to generate posters and banners.

An enthusiastic 93% of the teachers were interested in learning more about computers. Based on the results of the survey, a reference manual on computer applications for home economics teachers was written, a library of computer software was established, and inservice workshops were offered to home economics teachers.

The reference manual, Introduction to Computers for Home Economics Teachers, described the components and peripherals of a computer, common computer terminology, software selection and use, operating a computer, and simple programming. The manual was developed on a MacIntosh computer and printed on a laser printer to illustrate desktop publishing and the versatility of different computers and printers.

The software library included courseware for each content area of home economics. The library also included word processing, data management, and financial management software.

Six computer workshops were offered for home economics teachers. Sixty-five educators participated in the Saturday workshops. Apple IIe, Apple MacIntosh, and IBM microcomputers were used for instruction. Each participant worked individually on the computers. The course was team-taught by three instructors.

The educators were involved in a role-play lesson about how a computer operates. This was followed by examining computer hardware and peripherals. The educators took a CPU apart to examine the internal boards and followed each cable to the peripheral equipment. They also dissected a floppy disk and discussed how information is stored. These activities were followed with a discussion on caring for a computer and hardware purchases.

A lesson on software started with the input of a program using BASIC followed by a discussion of how software is developed. Guidelines for software selection and evaluation were presented.

Educators experimented with print shop and an variety of CAI software. They were instructed in the use of Report Card, a grading program, and Word Star, a word processing program. Each teacher developed a printed grade sheet as well as a document which was corrected for spelling errors using Proofreader.

The workshops participants were asked to rate each activity on its usefulness.

The role-play lesson on computer operation, examination of hardware and software, and discussion on caring for computers was found useful by 95% of the teachers. Eighty-five percent of the teachers found the tips for buying a computer, software evaluation, and the simple BASIC lesson beneficial.

All of the teachers rated the activity with Print Shop useful. Seventy-five percent found the time to experiment with CAI useful. Many participants commented that more time was necessary to maximize the value of this experience.

All of the educators rated the word processing/text correction lesson useful. Ninety-five percent found the grading program experience useful.

One hundred percent of the participants indicated that they had experienced an improvement in their attitude about computers. Eighty-five percent felt the workshop had decreased their "computer-anxiety" level. The educators indicated that the best features of the workshop were the hands-on experience, one-to-one attention given to participants, and the opportunity to try a variety of software and hardware. Suggestions for future workshops were to provide more time for experimentation with software and write lessons for classroom use.

When asked what software they would purchase for $100.00, most teachers listed word processing and grading programs. Some educators added Print Shop and CAI software.

The results of the initial questionnaire and the workshop evaluation make it possible to conclude that Hawaii home economics educators possess an enthusiastic attitude about computer use. These teachers use computers to help lighten their workload and enhance instruction.

Cecelia Thompson
University of Arkansas
212 Home Economics Bldg.
Fayetteville, AR 72701
(501) 575-4308

WHAT CAN I DO WITH A VIDEODISC?

Clo Wiltse, clo's----line Software, Inc.

ABSTRACT

The basics on videodisc applications. Delve into the multi-uses for videodisc slide and motion sequence collections. Learn how to use the videodisc player for simple slide presentations, programmed slide shows, and types of applications that may be made for education/training when the user interacts with a computer program which selects appropriate visuals for their responses.

PRESENTATION

Videodiscs were made for Home Economists! Of course I must acknowledge that they were also made for all educators, but there seems to be an edge to favor home economists since if we aren't dealing in life skills that beg for step-by-step analysis, then we're deep into over-all observation and analysis. Let me give you a couple of examples:
(Detailed process--fluted mushroom.
Overall observation--job interview.)

Movies, film strips, slide presentations w/their accompanying audio-cassette, hands-on experiences--all have helped our students "see" the things we talk about and read about in class. Yet the possibilities with laserdiscs promise so much more--because of the ability for us to INTERACT. Let's back-track a little.

Laserdiscs or videodiscs (terms used interchangably) are these beautiful psychedelic discs upon which images are imbedded. Also one or two audio tracks may be imbedded on the disc. Our laserdisc players are able to "read" these images and project them to us on our video monitor or TV set. The images are imbedded in spiral form starting at the center of the disc, winding their narrow way to the 54,000th "circle" at the outer edge. Laserdiscs come in two forms--CAV & CLV.

CAV is Constant Angular Velocity. Each image occupies an entire revolution of the spiral from the smaller "circle" in the center to the largest one at the outer edge. This gives each frame a specific address--the laser player, when commanded, can select the EXACT frame (out of a possible 54,000!) Remarkably, the time needed to search between the most distant locations takes anywhere from 8 to 34 seconds, depending upon the laserdisc player used.

CLV--Constant Linear Velocity, is commonly known as extended play. Every image is of equal length, and occupies its own space on the spiral. Thus, while only a few frames would reside on one of the inner spirals, by the time you get to the outer edge, that spiral contains many frames. This is why CLV cannot access a particular frame. It can be used to access by "Chapters" if the disc has been set up that way. But that is the nearest that may be available to specific frame or motion sequence recall. Therefore, for our educational interactive needs it has little value. It would have the same value as a (linear) movie and possibly be of a tad more value if it had Chapter stops.

Today I'm addressing my applications and advice to the classroom teacher and those who help them both in preparation and in-service training and support. FOR THERE EXIST TODAY MANY VALUABLE APPLICATIONS which we may implement to enhance our students' learning experiences.

Let's turn our attention to the truly interactive medium--CAV videodiscs. There are several levels of interactivity--two of them (Level 1 and 3) have practical, easy-to-use applications for us today:

Level 1:
Using laserdisc player and TV only

Level 2:
Using a laserdisc which has the program to run the interactive lesson embedded onto the laserdisc. This requires a special player and considerable more expertise and expense in producing the specific-oriented disc.

Level 3:
Using a computer to control the laserdisc player by telling it which frames or motion sequences to show, dependent upon the students' choices when running the computer program. This lends itself to the utilization of archival discs or the "repurposing" of discs. Thanks to simplified authoring systems (which I recommend for your initial experiences) one can set up a slide presentation or class activity during a prep period or in the evening before a class session.

Level 4:
Using other sophisticated equipment such as touch screens and voice synthesizers in addition to those mentioned above.

As a former classroom teacher it seems to me that there are four deterents to the implementation of new technologies in the classroom. The first is knowledge of its availability and applications. The second, the intimidation that many of us experience. The third and fourth are the cost and time factors that often tie our hands even if the first two are not factors. I'm assuming that those of us present have crossed over the first two hurdles. I think we must all help our cohorts and others to do so. Now let's turn to the last two hurdles. I'm here with good news!

Addressing the cost factor--Let me tell you what you'll need, and about my system. I got into it for under $1000, since I already owned a computer.

1. A personal computer.
   (mine's an Apple IIGS
   (a IIe or II+ is fine too
   (but a IIc would take more expensive interface set-ups.
2. A laserdisc player.
   (mine's a Pioneer LD-V2000 ($590)
   (get a player that has Level 1 & 3 capabilities
3. An interface to connect computer. to laserdisc player
   (mine's a VA I IIe ($150)
   (some players have RS 232 ports for computer connections
4. An authoring system.
   (mine's LaserWorks ($150)
   (there are systems with all manner of sophiscation and complication, and in prices ranging into the $2000 figure

The time factor coupled with cost influenced my choice of authoring system. I wanted one that would be simple to learn and quick and easy to use. I found it in LaserWorks. Agreed that it doesn't have all the features that might be desired, but it is it's very simplicity that allows you to use it with speed and ease. It's something you'll use rather than put on your shelf until you "have time for it."

There are equally easy, inexpensive authoring programs for other computer hardware systems. For instance, Vidkit is great for the IBM hardware.

Now that we've seen the system can be reasonably obtained, let's look at how we would use it:

LEVEL 1

Teacher/class presentations:

1. Slide presentation--slides won't spill, are all right-side up and facing front! You can keep one visual on the screen indefinitely without damaging disc or player as one would film products.
2. Enhanced lesson--in presenting information to the class, access the videodisc when slides or motion sequences would help. You can re-run portions, show each motion step-by-step, or slow up the action. Based on class discussion or needs, you can seek other visuals in an instant to illustrate a point, answer a question, evoke student thought.
3. Class, with teacher, follows skill procedure through step by step, allowing class to re-view a step or series of steps, and proceding as class needs dictate. Having the visuals on a screen that all can see enables all students to see details, rather than only the ones who are close to the demonstrator and thus have a preferred vantage point.
4. To help students learn effective personality skills, students could each be assigned a specific person to observe in a motion sequence, noting personality characteristics, mannerisms, etc., that enhanced their purpose as well as those that hindered their personal relationships or goals desired.
   a. This could be used in situations where role-playing might be used, to help the student "feel what it's like to walk in other's shoes".
   b. It's an ideal technique for study of job applications, business ettiquette, good work attitudes.
5. Promote observational skills by showing motion sequence, asking for recall of situation, physical environment, etc.
6. Class could observe video, then discuss their reactions with group.

7. Ideal tool for individualized learning. Include viewing of videodisc visuals as part of the required activities in each learning packet.
8. Students could be encouraged to incorporate visuals in reports to the class.

Student research:

(Request students to "browse"through videodiscs or a specific disc to search for items that would provide supplemental experiences or applications for a lesson. They must find specific examples, and present the scenarios and justifications for their choices.)

1. Find examples of how attitudes toward families have changed as reflected in art.
2. Find examples of how attitudes toward females have changed as reflected in art.
3. Find examples of how attitudes toward children have changed as reflected in art.
4. Observe fashion of the ages.
5. Find fashions that were truly flattering to the individual, regardless of the time period.
6. Find fashions/fashion periods that were comfortable, uncomfortable.
7. Find examples of effective human relationships.
8. Find examples of ineffective/poor human relationships.
9. Find examples of wise use of energy.
10. Find examples of food attractively served.
11. Find examples of attractive furniture that would also be comfortable.
12. Find examples of furniture arranged attractively and conveniently.
13. Find examples of good/poor safety practices--kitchen, home, child care, etc.
14. Find examples of how self-concept and/or our attitudes are reflected in appearance and performance and others' perception of us.

LEVEL 3

1. Slide presentations--have the preselected slides entered into a simple program so that all you have to do is press a key to advance to the next one rather than punching out the numbers of each frame on a remote control (and making mistakes!) Face the computer screen toward you, with your notes programmed on the screen, face TV or video monitor toward class to "show" slides or video clips for lesson.
2. Use for anything you might have used computer programs for--only now you have super visuals to boot!
   a. Excellent for simulations.
   b. Can be used for tutorials.
   c. Use for pre and post-testing.
3. Students can develop own program to "report" to others on their research.
4. Simulations are much more realistic--great for training and testing.
5. Students or student teams develop surrealistic video clips from bits and pieces to illustrate or create a mood.

What videodiscs are available for Home Economists to use? Many! There are discs already out there that will meet needs in all of our many disciplines. Archival discs such as our own Images: Individuals, Families, Society disc which our SIG developed this past year, the Helen L. Allen Textile Collection, and the USDA IVEN ShareDisc #1 are all collections of visuals that can be utilized to create any number and variety of lessons to meet one's individual needs.

Then, thanks to the possibility of "repurposing" (that is, taking an existing disk created for another purpose and finding applications in our own discipline--a skill we have been using creatively in all mediums from time immemorial) we have many others at our disposal. I found in the Superfight disk (CLV one side, CAV on the other) excellent examples of how one's self-concept affects one's actions and other's perception of us. (One can predict by watching the boxers in their corners how the next round will probably "go"!)

Here's a list of some I've found useful. They range in price from $25 to $250:

Helen L. Allen Textile Collection
The Cell Biology Videodisc
The First National Kidisc
Groller's KnowledgeDisc
The History Disquiz
Images:Individuals, Families, Society
Life Cycles
The Master Cooking Course
National Gallery of Art
USDA IVEN ShareDisc #1

What can you do with a videodisc? You can have a ball! You can add excitement to class, authenticity to lessons, encourage student response, develop creativity skills in students. With not too much money and little time you will find the videodisc can become your partner in helping students learn!

Clo Wiltse
Box 72
Volborg, MT 59351
406/784-2280

HUMAN RELATED FACTORS IN IMPLEMENTING A MANAGEMENT INFORMATION SYSTEM

Patricia V. D'Souza

California State University, San Bernardino

ABSTRACT

Virtually all industries use some type of management information system or decision support system (MIS/DSS). Unfortunately, many of the intrinsic benefits are not realized due to poor planning and implementation strategies. The lack of integration between human related factors and strategic business implementation processes is slowly being recognized as one of the determining factors in the success or failure of a system. This paper outlines five human related factors and problems that must be addressed in the implementation of a Management Information System.

INTRODUCTION

Virtually all industries use some type of Management Information System/Decision Support System (MIS./DSS). Unfortunately, many of the intrinsic benefits are not realized due to poor planning and implementation strategies. A great deal of literature focuses on the technical aspects of information systems implementation, while very little literature exists on the strategies applied to the interface of people with their machines.

The lack of integration between human related factors and strategic business implementation processes is slowly being recognized as one of the determining factor for the success or failure of a system. The issue of accommodating the human elements as we make the transition into the information/computer age is a complex one. This paper outlines five human related factors and problems that must be addressed in the implementation of a Management Information System.

I. GET MANAGEMENT INVOLVED

The need for "management commitment" at all levels is well established in the MIS implementation literature. Management must be involved in the implementation process for the following reasons.

1. First, subordinates will take a greater interest in the success of the project when their superiors show an interest in the outcome.
2. Second, the capital resources required for system implementation are substantial, and management must approve these expenditures.
3. Lastly, highly placed management support helps in crossing organizational boundaries and in restructuring activities.

There are varying degrees of management commitment. It is important to define what level of commitment is necessary to successfully complete the project. Effective management commitment involves (1) demonstrating a strong expression of support, (2) a willingness to allocate funds and staff, (3) taking an active leadership role, and (4) investing time, prestige, and attention to the implementation efforts (Bronsema and Keen, 1983).

II. ASCERTAIN THAT THERE IS A FELT NEED FOR THE SYSTEM

The system designer must analyze the user environment to determine the felt need for the proposed system. This analysis should not be restricted to the people proposing the system but should extend to the ultimate end users and those who must give inputs to the system (Lucas, 1982). Described below are the key factors in creating a need for a system.

1. Clearly identify the problem that the new system will address. Most problems are identified by the information user. If a problem is clearly identified, different departments may recognize the need for the system, thereby, the system will gain more broad based support throughout the organization.
2. Show how the system can help. This is perhaps the most important part of selling the system. A clear cut concise explanation as to how the system can help should be given in non-technical terms. The explanation should address areas of management's concern (e.g. return on investment, improved efficiency, improved effectiveness, increased productivity, etc.).

III. GET USER INVOLVEMENT

The users must be actively involved from the initiation of the system project through the life cycle (Ein-Dor and Segev, 1981). Research in human behavior indicates that people react poorly to change. Studies have shown that humans view change with a mixture of fear, ambivalence, lack of interest, and downright unwillingness to modify their behavior (Finn, 1983).

One of the major obstacles to the success of any system implementation is fear — fear caused by imagined threats, fears of the unknown, and fear of the complex. Analyst can help to curtail some of these apprehensions by spending more time with users and giving them more opportunity for participation in designing input, output, file contents, processing logic, etc. People resist change because they fear they will not be able to develop the new skills and behavior required. The analyst should discuss extensively with users the changes made in procedures and operations as they relate to the new system.

Many users are noncommittal to a new system because they do not know what it can do for them. Involving the user in the initial planning stage, explaining the advantages of the new system over the present system, and demonstrating how the system will benefit the user are primary ingredients in dispelling the fears and uncertainties encountered during the implementation process.

IV. PROVIDE TRAINING AND EDUCATION

Fear can also be reduced by well developed and thought-out indoctrination and training programs. Many studies in systems implementation indicate that the likelihood of success in an implementation effort increases substantially if there is a strong commitment to use education to lead the implementation process.

Many systems fail because of inadequate training. Training does not merely mean teaching a user how to input data into the system or how to get answers from the system. Training meaning education the user in the overall purpose of the system — what is it suppose to do, how does it do it, why do we want or need it done, and who must do it. Proper training does not just tell the users how important their contributions are, it lets them see that they are important integral part of the entire process and that the success of the system depends on how well they do their job.

Training should be provided for all personnel associated with the new system. Listed below are factors that should be considered when training users on the new system.

1. Training levels: The best approach at the outset is to teach just enough so that the user can begin to make practical use of the system. As experience build up, move to the next level.
2. Complexity: Move from simple to complex (e.g. how to turn equipment on and off, how to sign on and off, how to access a network, use an electronic spreadsheet, data management system, etc.).
3. Time: The time factor depends on the level of user knowledge, sophistication, and involvement in the information systems area. Users with little or no background will obviously take longer to become productive on the new system than an experienced user.
4. Instruction: There are many different methods of instruction used in training today. Two of the most common methods are human instruction and software. The analyst must determine which method or combination of methods are most effective for the user in that particular environment.

V. CONSIDER USERS' ATTITUDES

MIS can and does fail where user psychological reactions and organizational factors are ignored by system designers. Ultimately, the success of any system will greatly depend on users' attitudes toward the system. Acceptance of a system will be reflect in interaction between the type of user and the kind of thing the user is trying to do. Some of the critical factors that influence user attitude and acceptance of the system are: (1) user friendliness, (2) ease of use, and (3) ease of learning.

The designer must therefore know the personnel who are going to use the system. If the users are highly technical people, then ease of use is not a major requirement. However, if the users are to be clerks and other regular office employees, then user-friendliness of the system should be a major consideration.

It is vital to the success of the system to insure that users have positive attitudes toward the system. Listed below are ways management can influence users' attitudes.

1. Establish effective communication channels between user, analyst, and management. The follow-up/feedback approach — making communication a two-way street should be emphasized.
2. Establish a reward system that will motivate the user. Rewards or goal accomplishments can result in increased user satisfaction and an increased chance of system success.
3. Keep interface simple. The appeal of simplicity is quite direct: simple solutions are easier to understand, easier to implement, and usually easier to control and modify than complicated solutions.

CONCLUSION

As computer technology proliferates in our society, the user-system interface and the psychological aspects involved in the implementation of information systems are being recognized. The intrinsic benefits of a MIS can only be realized if human related factors are considered during the planning stages of the implementation process. The five human related factors described in this article may be valuable elements in the design and delivery of successful management information systems.

REFERENCES

Bronsema, G.S. and Peter Keen, "Education Intervention and Implementation in MIS," Sloan Management Review, Summer, 1983.

Ein-Dor and Segev, A Paradigm for Management Information Systems, Praeger, New York, 1981.

Finn, Nancy B. "Interfacing People with their Machines," National Computer Conference Proceedings, Anaheim, CA, 1983.

Lucas, Henry C. Information Systems Concepts for Management, McGraw-Hill, New York, 1982.

Patricia V. D'Souza
Department of Decision Sciences
California State University, San Bernardino
5500 University Parkway
San Bernardino, California 92407

TEST DESIGN FOR COMPUTER MANAGED INSTRUCTION SYSTEMS

by
Robert L. Jorczak
Franklin C. Roberts

Control Data Corporation

ABSTRACT

This paper reviews the strategies and components of testing; providing operational definitions of group, individual, and adaptive tests. It is proposed that these diverse testing strategies share common components, and that all strategies can be genereated by specifying the parameters of these components. These six components are: item selection, item presentation, test length, test termination, scoring, and student initiated options. By providing a facility for specifying the options in each of these categories, a CMI testing system can be easily expanded to include a wide range of testing alternatives.

INTRODUCTION

Today's microcomputers and local area networks provide sufficient computing resources to accomplish comprehensive computer-managed instruction (CMI) functionality. Such automated management systems perform four major tasks: 1) curriculum definition and organization, 2) assessment of student capabilities and needs, 3) assignment of appropriate learning resources, and 4) reporting of student status and performance.

A key function to the success of any management system is the accuracy of the assessment component. If student needs are not accurately assessed, then a student either spends time studying and retesting material already mastered, or worse, unmastered material is not identified and addressed. It is important that CMI system designers provide flexible and comprehensive assessment testing capabilities in the systems they design, and that courseware designers properly use such capabilities to create accurate, effective, and efficient tests.

Fortunately, designers of computer-based testing utilities can refer to a large body of research data on assessment testing in deciding the goals and structure of a system. Much is known about how to perform efficient and accurate assessment testing of both the criterion-referenced and norm-referenced variety. These data and conclusions provide a firm basis for defining which testing strategies and options are optimal for various assessment situations.

Accordingly, the initial task in designing a CMI system is to identify which testing strategies must be supported by the system. If the system is intended to support specific testing situations, this number may be small. If the system is intended to support a diverse range of testing situations, the number of required strategies may be quite large. This paper will consider the issues surrounding the design of a system which supports a diverse number of testing strategies. This means that the designer is faced with another task: identifying common characteristics and needs most of the required testing strategies. An analysis of such characteristics is examined in this paper.

In addition to the identification of common, basic characteristics of diverse testing strategies, a testing system designer must also consider what information needs to be stored by the system to support the target testing strategies. A related task is the identification of data that is required to perform various test and test item analyses. Test developers need data about the performance of their tests and test items.

Designing for micro-based systems places another requirement on system designers. Micro systems, even though their capabilities have increased dramatically in recent years, still have limited resources compared to large mainframe computers. CMI systems tend to strain current, micro system resources to their limit in storing test items, test data, and analysis data. This means the testing system designer must carefully identify what data is minimally necessary to support target testing strategies and the most common and useful analyses.

TESTING FEATURES

An educational test is a series of questions or exercises for measuring the skill, knowledge, capacities, or aptitudes of an individual or group. While all tests share this goal, there are many ways in which this goal is accomplished. These various strategies are effected by the purpose of the test, whether it is delivered individually or by group, the degree of flexibility in the test delivery medium, the data analysis techniques required, and so on.

Most CMI systems are designed to be used for individualized instruction in a computer based environment. While this puts some constraints on the applicable testing strategies (such as the difficulty of grading essay questions), the flexibility of such systems generally allows the delivery of a wide range of testing strategies. The following sections describe some of the basic issues in test design and what the requirements of the corresponding strategies are.

The basic issues of testing which are described here are organized into four categories: 1) purposes of tests, 2) test items, 3) test delivery, and 4) test scoring.

Purposes of Tests

This section describes two basic categories of test purposes: norm-referenced and criterion-referenced. The distinctive feature of norm-referenced measurement is that scores are interpreted in relation to some appropriate group of examinees. For example, a score which places an individual at the 70th percentile of all graduating high school seniors is an example of a norm-referenced measure. It is considered to be norm-referenced in that the individual's score reflects his or her performance in direct relation to the performance of all other persons who took that test. The key issue in norm-referenced testing is the ability to be able to discriminate between individuals within some group.

Thus, norm-referenced menasurement seeks to relate an individual's performance to others of an appropriate group. A norm-referenced test (NRT) is a test that is used for such purposes. While norm-referenced tests provide useful information, they offer little help in defining what a student knows; only how much that student knows in relation to others. The need to have some absolute measure of what a student knows spawned the concept of criterion-referenced measurement.

The development of criterion-referenced testing technology parallels a change which has occured in our conception of achievement. Traditionally, educational practice in the schools has held the amount of time allowed for learning fixed, while the degree of achievement was allowed to vary. In this newer view of achievement, however, the role of time and achievement is reversed: achievement, defined as master, is held constant for all students, while the amount of time allotted to each student is allowed to vary so that each student is given the necessary time to achieve mastery (hence the term mastery learning).

This fixed degree of achievement is often referred to as a domain, or a domain of behavior. Thus, a domain-referenced test is one that results in a measure which clearly reflects some well defined amount of achievement. Implicit in this model is the notion of mastery - at some point, according to some criteria, a student will be defined as a master. This is usually based on a test result, where some mastery criterion is used to make decisions between masters and non-masters. A test that is used in this way is referred to as a criterion-referenced test (CRT).

Thus,there are two broad categories of purposes for which tests are used: norm-referenced and criterion-referenced. These different purposes will effect how the test is built, how the items are chosen, when the test will be terminated, and so on. It should also be mentioned that in addition to these broad categories of test purpose, tests can have more specific purposes.

For instance, there are three basic purposes for which pretests can be used. 1) Readiness: assessing whether or not a student has the prerequisite skills and knowledge necessary to learn the current material. 2) Assessment: determining a student's overall level of understanding of the topic to be taught. 3) Sensitivity: a pretest can be used as an advanced organizer, senstizing a student to the types of questions and related content she or he will encounter. A pretest may be used for more than one of these reasons, resulting in six different pretest purpose conditions. Each of these pretest purposes effects various test delivery parameters, such as selection of items, the order of items, and test termination.

Test Items

The test item is the basic component of all tests. In terms of the issues involved in designing a CMI system, there are two relevant aspects of test items: item characteristics and item selection techniques.

Item characteristics define the relation between an item and some external measure. There are two such external measures: content, and student data. Items are often related to content by their association with a particular objective. This content-based association is the basis of all criterion-referenced tests. If a test covers only one objective, the association is implicit. However, a pool of items may cover many objectives, and it will often be important to know which objective an item reflects. This characteristic can be stored directly with the item or it may be implicitly defined by the organization of the items in the pool.

Items can be related to student data in many ways. One of the most frequently used item characteristics of this type is item difficulty. Item difficulty is a measure reflecting how many students got the answer correct. Another item characteristic based on student data is item discrimination. This measure reflects how well the item discriminates between students who do well on the test, versus those who do not. For instance, an item discrimination value might point out an item which is answered correctly only by people who did not pass the test. Item characteristics which are keyed to student data are useful in both norm- and criterion-referenced tests.

Item selection techniques are used to choose the test items which a student will see. Item selection strategies range from the very simple, such as "Choose all items," to the extremely complex. T power of computer-based testing is in the ability to immediately analyze a student's response, and then determine what action to take next. Such tests are called adaptive tests, and are described in the next section. What allows these tests to be adaptive is their ability to choose test items on more than just a random basis. All advanced item selection techniques are based on the features of item characteristics, as described above.

Testing Strategies

There are a wide variety of possible testing strategies, but for purposes of this paper they are organized in three categories: 1) group based testing, 2) individualized testing, and 3) adaptive testing. Most of the conditions for these strategies have already been described. Group based testing is a strategy that presents all students the same test, usually at the same time. While group based tests can be delivered via computer, they are most generally paper based tests.

Individualized tests are delivered to one student at a time, and are generally designed to meet a student's particular needs. Many individualized tests are based on content characteristics, ensuring that a student is getting a mastery test on the material which the student has just studied. While item characteristics are a key issue upon which individualized tests are designed, it is important to note that the items for these tests are chosen *before* the test is started. While a test may be terminated before all items are presented, the key issue here is that the selection of items is not based on student performance during the test.

The third category of test here is called adaptive testing. An adaptive test is an individualized test that chooses items based on a student's performance *during* a test. For instance, consider a pretest that covers a large curriculum, where the purpose is to place the student in the proper place in that course of study. If the content is hierarchical and later topics build on earlier ones, the first few items might be from the middle part of the curriculum. If the student does not get them correct, items might then be chosen from the first part of the curriculum.

A similar strategy could also be used with item difficulty. In this way, the test adapts to a student's performance during the test. This type of test is called an up-and-down adaptive test, as it presents items which are either more or less difficult than the last item given. The difference in difficulty between the last item and the next item is referred to as the step size.

Many adaptive testing strategies are possible, and the term is also often used by others to define certain categories of tests which we have labeled individualized. A new set of techniques have evolved in this area based on Item Response Theory (IRT) (Lord, 1978). Adaptive tests are extremely sensitive to student performance and provide the most information. Adaptive tests generally reduce the amount of time required to assess an individual's level of competence, and increase the accuracy of the measurment.

Test Scoring

The final aspect of testing to be discussed concerns test and test item scoring. There are two basic issues to be discussed: item statistics and test statistics. These two types of statistics are important in the design of CMI systems as they relate to the types of data to be stored, the item characteristics which are based on student performance, and the general validity and reliability of the overall testing systems.

Item statistics are related to the item characteristics based on student performance described above. An item statistic is any measure of student performance on a particular item. These statistics are used for item analysis as well as item selection. A range of statistics have been proposed, both for items used in norm- and criterion-referenced tests. In CRTs, item discrimination indices reflect the performance on an item of those who master the test and those who do not. Differences which are found between these groups are presumed to be the result of the instruction. Accordingly, these statistics are often referred to as instructional sensitivity indices (Haladayna & Roid, 1981; Berk, 1981a).

Test statistics are used to measure the overall effect of a test, as opposed to the effect of a single item. Common test statistics include the mean and standard deviation, the reliability estimate, and measures of test validity. Like the test item statistics, there are a variety of approaches for measuring many of these test statistics.

TEST PARAMETERS

As described above, there are a variety of testing strategies and purposes. Analysis of these various testing alternatives indicate that they all share a common set of basic functions. The design of any testing utility must consider how these basic functions will operate, and how they might be altered by test makers.

All testing strategies must specify how the following six tasks are accomplished:

- Test Item Selection
- Test Item Presentation
- Test Length
- Test Termination
- Test Results (Scoring)
- Student Initiated Options

Test Item Selection

The test item selection task specifies how items are chosen from a pool of potential items for use in a specific test. There are three main categories of item selection specifications: 1) select all items in the pool, 2) randomly select items from the pool, and 3) select items based on item characteristics. The "select all" strategy is used most often in simple group tests used in classrooms. This is generally the least efficient strategy in achieving an accurate assessment of a student's capabilities.

When the number of items in the pool is greater than the number of items to be presented, additional selection specifications are possible. One of the simplest is to randomly choose a subset of items from the pool. This allows the creation of tests with different items on the same subject. This offers advantages in test security and therefore results in increased assessment reliability.

Item characteristics are central to item selection. The most important characteristic of any item is the appropriateness of the item to the goal or objective of the test. Often this is implicit in the organization of an item pool (e.g., by learning objective) or is implied merely by inclusion of an item in a pool that addresses a certain content area. Advanced testing pools may explicitly store a range of such information with each item.

Other item characteristics can be useful for item selection. Selection may be based on item characteristics such as item difficulty or reading level. Some selection strategies may be based on student performance data such as whether or not a particular item or set of items was previously presented.

Test Item Presentation

Item presentation is another basic parameter of test design. Given that item selection specifies a subset of test items from a pool that will be included in a test, item presentation specifies in what order the selected items will be presented.

Selection and presentation specifications can be confused, as selection specifications may include a selection order that is also intended as the presentation order. For example, items for a test may be selected randomly from a pool and then presented in that same random order to students. In a sense, these items are presented sequentially, as defined by the random selection.

A key distinction is that selection can be done prior to testing, whereas presentation order is, by definition, done during testing. This is potentially confusing when the selection specification involves decisions that can only be resolved during testing (e.g., in adaptive testing).

Item presentation alternatives are often limited to sequential (i.e., as stored), random, or "as selected." Adaptive testing strategies, as mentioned above, involve complex presentation specifications based on selection involving item difficulty and the student's preformance on previous items.

Test Length

An important issue in creating any test is the number of test items required to meet the testing purpose. Berk identified four factors to consider in determining test length: "1) importance and type of decision to be made with test results; 2) importance and emphasis assigned to the objectives; 3) number of objectives; and 4) practical constraints" (1981b).

Roberts (1982) discusses how, in CRTs, test designers seek to minimize the incidence of erroneous mastery and non-mastery assessments. Obviously, longer tests will make fewer errors than shorter ones, and some techniques for choosing good test lengths are known.

As with most basic testing parameters, test length is related to other parameters. For example, the actual number of items presented may not be the maximum test length if early termination conditions are included in the testing strategy. The length of an adaptive tests is dependent on the selection/presentation algorithn and number of calibrated items available.

Test Termination

A testing strategy may include early termination conditions that address testing efficiency by terminating a test before the expected number of items have been presented. Examples of such conditions include strategies where testing is terminated when it is clear that the student cannot meet the test mastery criterion even if all ramaining items are answered correctly. Early termination could also be invoked if the mastery criterion is met before the maximum number of items have been presented. Another example is when the test has a time limit.

Test Result (Scoring and Data Storage)

As a basic test parameter, test result has the most alternatives. Even so, the main goal of all test scoring procedures is an evaluation of student performance in regard to the testing goals or purposes. A CRT assesses student status in terms of mastery of learning objectives. A NRT assesses student status in terms of a particular population of students. Both express the assessment by the test result.

Most test results are expressed in terms of a quantity. A test result may be expressed as an integer value (score), a mastery/non-mastery judgement, or less common evaluation quantifiers. Even if simple mastery evaluation is indicated, many alternatives for specifying mastery criteria are possible. The mastery criterion may be based on the number of items answered correctly, point totals amassed from the different items (which may be weighted), or more arcane scoring procedures which may be associated with the item presentation strategy. Integer scoring offers even more opportunity for variations in scoring procedure.

Besides reporting a score representing student performance, test results may collect and store data for various other purposes including: 1) support of testing strategies, 2) student performance reports, and 3) test and test item analysis and evaluation.

Compute -based testing utilities must store various data to support testing strategies. Test item characteristics must be available in order to perform advanced selection strategies. For example, most adaptive strategies must know item difficulty.

Testing utilities also store data for student performance reports. Data is also stored in order to perform various analyses of tests and test items including difficulty (P-values), item discrimination, test effectiveness and reliability.

A system designer must consider the requirements for these functions in defining a set of stored data. This is a particularly important task if the system is micro computer-based as system memory resources are more limited than large mainframe computers. Accordingly, it is important to identify the minimal set of data required to meet the needs of these three functions.

We propose a minimal, but flexible set of data to be used in analysis, as shown below:

I. For each item presented:
   A. The number of initial presentations to all students.
   B. The number of times answered correctly by students who:
      1. mastered the test
      2. did not master the test
      3. are taking the test for the first time
      4. are taking a retest
   C. Frequency counts of wrong answers:
      1. of each distractor (selected response items)
      2. of anticipated and unanticipated answers
II. For the entire test:
   A. The number of tests presented to all students.
   B. The number of tests mastered.
   C. The number of students who master on first try.
   D. The number of students who did not master first try.
   E. The number of students who did not master second try.
   F. The sum of scores for first try.
   G. The sum of scores squared on first try.

Student Initiated Options

Many testing strategies include capabilities that may be optionally invoked by the student taking the test. These include the ability to skip an item and have it presented after all items have been presented. Other options include withholding test scoring until the student has answered all items, and has been given the opportunity to change his or her answers. Some tests may call for student access to special reference information or data, or a statement of the test objective. Many of these options are incompatible with certain item presentation, termination, and scoring specifications but a comprehensive system should support such options where appropriate.

An interesting aspect of student initiated options is that they are generally available with paper-based tests but frequently not found with computer-based testing. Future CBE test designers need to consider including such options to support a wider range of testing strategies.

Example Specifications

A testing strategy is complete when the above parameters are specified. Three examples are provided below.

Parameter	Value
Item Selection:	Any item associated with objective
Item Presentation:	Sequential as listed in pool
Length:	10 items
Termination:	When test length is achieved
Result:	Mastery if 8 items answered correctly
Student Initiated Options:	Skip items, change answers

Figure 1: A group based testing strategy

Parameter	Value
Item Selection:	Randomly select
Item Presentation:	As selected
Length:	100 items
Termination:	Max length, or score of 80 is achieved
Result:	Integer score ofcorrect answers
Student Initiated Options:	Item skipping

Figure 2: An individualized testing strategy

Parameter	Value
Item Selection:	Calibrated items covering objective
Item Presentation:	Adaptive, Up-and-down, fixed step
Length:	Variable, up to max calibrated items
Termination:	When test result is established
Result:	Percentile rank
Student Initiated Options:	None

Figure 3: An adaptive testing strategy

RAMIFICATIONS TO CMI TEST DESIGN

This paper has proposed that the six parameters described above include all of the functions required to support the vast majority of alternative testing strategies. In building any test, or in designing any testing system, each of these parameters is defined, either explicitly or by default. By formalizing these parameters, any testing system can support a wide range of testing strategies. Accordingly, a designer of a general test building utility or testing system would only have to identify which parameter specifications will be supported by the system, and how the users would choose from these alternatives to implement their strategies.

This does not imply, however, that identification of test parameter specifications is a simple task. The above testing parameter specifications can effect each other in complex ways. Such effects must be understood to ensure that desired testing strategies are indeed accomplished without unexpected side affects. Many of these interacting effects are confined to a specific category of testing (i.e., group, individual, and adaptive).

For instance, while there are a variety of test termination alternatives, within the group testing strategy there is only one: no early termination. As another example, within the adaptive testing category, item presentation strategies are moot, as the item selection strategy determines which item will be presented next. Using these six parameters and three categories of tests, the key task of the test utility designer is to examine the interrelationships for the types of tests to be supported, and design the user interface accordingly.

BIBLIOGRAPHY

Berk, R. A. (1981a). Criterion-referenced measurement: The state of the art. Baltimore: The John Hopkins University press.

Berk, R. A. (1980b). Practical guidelines for determining the length of objectives-based, criterion-referenced tests. Educational Technology, V.20, No. 11, pp. 36-41.

Haladayna, T. & Roid, G. (1981). The role of instructional sensitivity in the empirical review of criterion-referenced test items. Journal of Educational Measurement, V. 18, No. 1, pp. 39-53.

Lord, F. M. (1978). Applications of Item Response Theory to Practical testing Problems. Hillsdale, N.J.: Erlbaum Press.

Roberts, F. C. (1982). Asummary of the literature on criterion-referenced testing and related issues. Minneapolis: Control Data Corporation, Educational Research Technical Report TR-1982-11.

Contact: Robert L. Jorczak
449 11th Avenue South
Minneapolis, MN 55407

(612) 827-8961

LADAR: a LAnguage for Describing Anticipated Responses in CAI answer judging

Hiroshi Kato, Yasuhiro Morimoto

C&C Information Technology Laboratories, NEC Corporation

ABSTRACT

A micro-computer language for judging a student's response in courseware development is described. It facilitates authors in describing various answering styles, such as strings, numerics, subrange of both, algebraic expressions, and any combination of them. Its concise description can flexibly identify a wide variety of response patterns. Besides, the syntax is easy enough for beginners to master due to its systematized structure.

1. INTRODUCTION

As answer judging is a main contact point between a student and a CAI system, it becomes an important part of the CAI man-machine interface.

Some CAI systems, especially such authoring languages as TUTOR (Sherwood, 1974, Avner, 1981), WISE (Fairweather et al., 1984), and PILOT (Apple Computer Inc., 1980), are adequate to judge student's responses, but these are often so arcane and syntax-rich that high programming skills would be required in order to take full advantage of the language. On the other hand, other systems are easy to use in answer judging, but are less efficient. Consequently, since it was considered to be necessary to accomplish both easy and highly efficient answer judging, LADAR has been developed.

The rest of this article reveals an outline of LADAR, the main functions that contributes to its efficiency, and some examples in using LADAR.

2. LADAR OUTLINE

2.1 LADAR Functions

LADAR is a LAnguage for Describing Anticipated Responses. The LADAR system is a functional module which interprets a LADAR statement representing anticipated responses, compares these recorded statements with a student's response, and judges whether they match or not.

In order to specify the LADAR capability, general functions, which are thought to be required for CAI answer judging, are discussed individually. They include:

1) Full-string or sub-string match
2) Judging numeric with/without a tolerance
3) Ignoring some part of student's response
4) Ignoring the order of elements
5) Judging algebraic expression
6) Integrating the above functions
7) Utilization ease
8) Productivity

Regarding the preceding 1) through 4), LADAR's performance is equivalent to that of the existing CAI systems, but the method taken in LADAR is unique and elegant. Details will be discussed later.

Regarding 5) judging algebraic expression, several authoring languages, for instance TUTOR and WISE, use a method in that a student's expression is evaluated and compared with the anticipated response by value. This method, however, cannot be applied to some kinds of algebraic problem, such as factorization and simplification, because the original expression given in the problem becomes the correct answer. On the other hand, LADAR makes syntactic matching, by which the form of an expression can be examined. Therefore, LADAR can be applied to any kind of algebraic problems.

Regarding 6) integrating the above functions, none of existing CAI systems allows unrestricted mixture of answering styles in a response; while it is allowed in TUTOR, but restricted to either a numeric and a set of units, or strings and simple arithmetic expressions including no variables. On the other hand, LADAR allows unrestricted integration of answering styles, such as strings, numerics, subrange of both types, and algebraic expressions.

Regarding 7) utilization ease, LADAR is composed of a small number of elements and element-combining rules. Its small syntax facilitates the author's ability to master it with little effort. In addition, a complex judging could be accomplished with the combination of simple rules. Therefore, even beginners, as well as experts, can take advantage of LADAR in their own way according to their skill.

Regarding 8) productivity, a variety of response patterns derived from a student's misconception can be described with a LADAR statement in principle. Consequently, a description written in LADAR will generally become more concise than that written in existing authoring languages, if the performances are the same.

2.2 Components

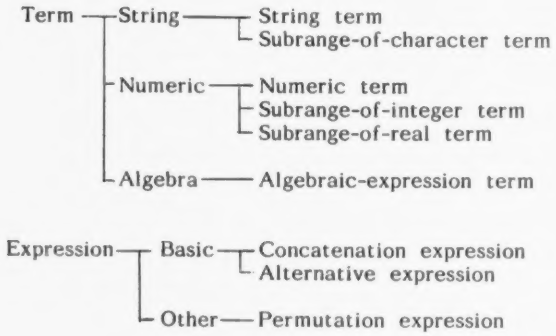


Fig. 1 LADAR Components

A LADAR statement is composed of either a term or an expression. As shown in Fig. 1, the terms are elements, such as strings ("a pen"), subrange of character ("a".."f"; any character from "a" to "f"), numerics ([3.14]), subrange of integer ([1..5]; any integer from 1 to 5), subrange of real ([1<<5]; any number x:1<x<5), and algebraic expressions ([2x+1]). The expressions are functional units consisting of terms and/or expressions, such as concatenation ([3]cm; numeric+string), alternative ((movie, picture, cinema); any one of these), and permutation (<x,y,z>; any one of xyz, xzy, .. , and zyx).

Seeing the syntax from the standpoint of symbolic logic, a term corresponds to a terminal symbol, and an expression corresponds to a nonterminal symbol. LADAR system regards a LADAR statement as a start symbol or an axiom, deduces some series of terms, and compares them with an actual student response.

2.3 Implementation

LADAR has been installed on the microcomputer NEC PC-9801 series loading NEC V30 CPU (Intel 8086 compatible), and written in Lattice C on MS-DOS. Its object size is only 38Kbyte, using a small model. Furthermore, it works so fast that most of an answer judging can be completed within a second.

It should be noted that LADAR is not a programming language to support the whole courseware development, but a functional module for the use in judging responses. The module accepts both a LADAR statement and a student response, and outputs an indication as to whether they match or not. Thus, it is loosely connected to main program. Consequently, it would be basically possible to incorporate LADAR into conventional authoring systems.

3. LADAR TEXT DESCRIPTION

The following fundamental elements play an important part in LADAR's syntax, because most of the other high level terms or expressions could be rewritten by using fundamental elements. However, the high level elements are practical and convenient in judging a certain style of answer. Among the high level elements, only algebraic-expression term and permutation expression will be mentioned here, due to space considerations.

3.1 LADAR Fundamentals

(1)String Answer (String Term)

In judging a simple string answer, the String Term is applicable. For example:

"triangle" ,or simply triangle

(If a string contains reserved symbols of LADAR, double quotes are essential.)

This example will exactly match the response 'triangle'. In order to make judgment more flexible, the following three kinds of wild-card symbols are applicable. Though LADAR can neither ignore specified words nor allow a spelling tolerance, as TUTOR or WISE can do, LADAR's wild-cards can still compensate for those functions:

_ An underscore is the metasymbol which will match null or greater length of spaces. For example, the LADAR statement "a_ pen" will match the responses 'a pen' and 'a   pen'.

? A question mark is the metasymbol which will match an any symbol. For example, the LADAR statement "b?c?cle" will match the responses 'bicycle', 'bycicle', and 'b cocle'.

* An asterisk is the metasymbol which will match null or greater length of any symbols. For example, the LADAR statement "*pear*" will match the responses 'I like pears.', 'My favorite fruits are pears', and 'spearmint' (which is a mistaken matching).

Besides, such attributes as capitalization and spaces between words may either be ignored or declared to be significant.

(2)Numeric Answer (Numeric Term)

In judging numerics, the Numeric Term is applicable. For example:

[30], [30.00], or [0.3e2]

These will equally match the responses '30', '30.0', or '3.0e1'.

Some of existing authoring languages can allow a tolerance of numerics, but LADAR's numeric term itself cannot do so. As a substitute for that limitation, other terms are applicable for tolerance; i.e., subrange-of-integer term and subrange-of-real term. Their details are, however, omitted due to limited space.

(3)Mixture of String and Numerics
(Concatenation Expression)

In judging the mixed response of strings and numerics, the Concatenation Expression is applicable. For example:

[30]cm

This will match the responses '30cm', '30.00cm', and '3.0e1cm'. Its syntax is only to concatenate terms and expressions in series.

(4)Enumeration of Anticipated Responses
(Alternative Expression)

In case that more than one response is acceptable or thought to be equivalent, the Alternative Expression is applicable. For example:

{[30]cm,[0.3]m}

This will match the responses '30cm', '0.3m', and '30.0cm'. The general form of its syntax is as follows:

{term/expression, ..., term/expression}

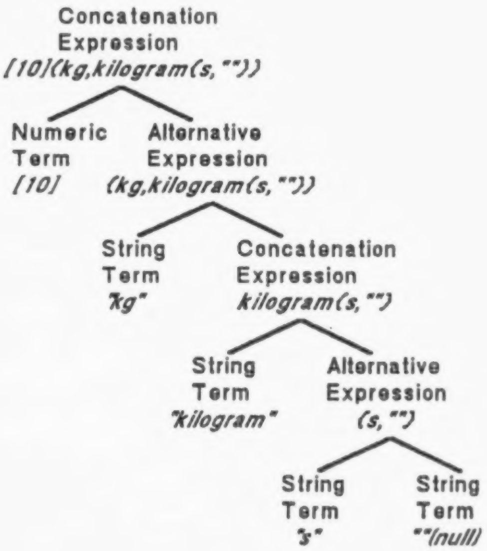


Fig. 2 Syntactic Structure of the LADAR Statement: *[10](kg,kilogram(s,""))*

(5)Integration

As the syntax indicates, an expression can include any expressions, as well as terms, which is a common feature for any other expressions. This feature realizes an unrestricted integration of answering style, which has never been accomplished before. The following is an example of nested expression:

[10]{kg,kilogram{s,""}}

(where, "" means null string)

The syntactic structure of this statement is shown in Fig. 2. This will match the responses '10kg', '10.0kilograms', and '10kilogram'.

TUTOR has a special statement to handle units, which is restricted to the answer of number+units style. Therefore, it cannot handle, for example, an answer regarding an algebraic-expression (which will be mentioned in the next item)+units style, such as '2x+1 kg', while LADAR can do it in such an easy way as [2x+1]kg.

3.2 Algebraic Expression Answer
(Algebraic-expression Term)

In judging an algebraic expression, the Algebraic-expression Term is applicable. For example:

[(x+3)y], [(x+3.0)*y], or [y(3+x)]

These will equally match the responses '(x+3.0)y', '(3+x)*y', 'y(x+3)', and '((x)+(3))(y)', but will not match the response 'xy+3y'. The system tolerates all the variation in expressions, in which the commutative law is applied to all the contained arithmetical operations; i.e., addition, subtraction, multiplication and division.

Some existing authoring languages can handle the expression by numerical evaluation. The following is a typical example in TUTOR:

calc x ← 1.2	(Substitute 1.2 for x.)
calc y ← 3.4	(Substitute 3.4 for y.)
arrow 1420	(Enter judging state.)
ansv (x+3)y	(Evaluate response and compare it with the value of (x+3)y, i.e., (1.2+3)*3.4=14.28)
no	(Terminate judging if not match.)

This method has a disadvantage in that it does not fit some problems, for instance factorization and simplification, as mentioned earlier.

In fact, previous Numeric Term is only a special form of this expression.

3.3 Permutation Expression

(1) Word Order Ignorance (Permutation Expression)

In judging the answer of multiple-choice question allowing more than one choice, the Permutation Expression is applicable. For example:

<a,c,d>

This will match the responses consist of 'a', 'c', and 'd', irrespective of the order. This example is, therefore, equivalent to the following alternative expression.

{acd,adc,cad,cda,dac,dca}

The general form of its syntax is as follows:

<term/expression, ..., term/expression>

In order to accomplish the same function using existing authoring language, troublesome enumeration of all the conceivable patterns might be necessitated. Following is an example in PILOT:

```
m:%acd%!%adc%!%cad%!%cda%!%dac%!%dca%
```

(where, % is the symbol for a 'required beginning or end of match text item', and ! is the 'or' symbol.)

(2) Delimiter Specification (/ Option)

In the above example, each alternative may be separated by a delimiter, such as a space or a comma. In order to specify delimiters, the / Option is applicable. Syntactically, it can be added at the end of the term/expression list in the permutation expression. In the following example, /{" _and_ ",",_"} is the /option.

<a,c,d/{" _and_ ",",_"}>

This will match the responses 'a,c,d', 'c, d, a' and 'a, d and c'

TUTOR has a mode to allow any word order, i.e., 'specs noorder', which has a restriction in that individual words must be separated by space or punctuation. The answer written in Japanese or Chinese cannot be judged under this restriction, because such languages do not leave a space between words.

4. UTILIZATION EXAMPLE

In the following examples, it is assumed that student's response will be compared with LADAR statements in serial order, and that a branch toward other frames will be made when they match.

4.1 Example 1

[Question]

Fill in the blanks with proper words so that the following two sentences become equivalent.

Juan compró la casa.

La casa (   ) (   ) por Juan.

[Judging]

1. _fue _comprada_
   (Correct answer. More than one space between words is acceptable.)

2. *comprado*
   (Incorrect gender. This will match the response containing a substring 'comprado'.)

3. _{es,era*,ser*,son} _compr*
   (Incorrect tense. This will match 'es comprada', 'sera comprada', et al. If the response is 'es comprado', a branch will not be made here, because this has been matched with previous LADAR statement "*comprado*".)

4. _fue _compr*
   (Incorrect inflection. This will match 'fue comprados', 'fue comprida', et al.)

4.2 Example 2

[Question]

Simplify the following expression.

6a+5b-4(a+b-0.5)

[Judging] --- assume the capitalization-ignoring mode.

1. [2a]+b+[2]
   (Correct answer; inclusive of the variable's alphabetical order. This will match '2a+b+2', '2.0a+b+2.0', et al.)

2. [2a+b+2]
   (Correct answer. This will match '2a+2+b', '2.0+b+2a', et al.)

3. [2a+6b-0.5]
   (Incorrect answer. Such a misconception as '-4(a+b-0.5)=-4a+b-0.5' might occur.)

4. {*ab*,*ba*}
   (Incorrect answer. The response is containing a substring either 'ab' or 'ba'. Such a misconception as '2a+b=2ab' might occur.)

5. CONCLUSION

The concept and usage of LADAR has been described. In conclusion, LADAR features are summarized as follows:

1) Its syntax is simple enough to learn easily, although judging capability is highly efficient.

2) Answering styles, such as strings, numerics, and algebraic expressions, are integrated.

3) Separators between words can be specified explicitly by author. This is for the sake of processing a language not leaving a space between words.

4) It works rapidly on micro-computers with a small memory capacity.

5) It has a high modularity, because it is loosely connected to the main program.

In the future, LADAR's capability of abstracting a response pattern will be expanded to a system in that various patterns of correct/incorrect responses can be produced from simple descriptions of student's misconception rules.

ACKNOWLEDGEMENTS

The authors wish to thank Takahiro Sato and Masahiko Kurata for their advice on this work.

REFERENCES

Apple Computer Inc. (1980). Apple PILOT Language Reference Manual.

Avner, E. (1981). Summary of TUTOR Commands and System Variables (tenth edition). PLATO Services Organization.

Fairweather, P.G. and O'Neal, A.F. (1984). The Impact of Advanced Authoring Systems on CAI Productivity. Journal of Computer-Based Instruction, Vol.11, No.3, pp.90-94.

Sherwood, B.A. (1974). The TUTOR Language. Computer-based Education Research Laboratory and Department of Physics University of Illinois, Urbana.

Contact: Hiroshi Kato
Information Basic Research Laboratory
C&C Information Technology Laboratories
1-1, Miyazaki 4-chome
Miyamae-ku, Kawasaki
Kanagawa 213, JAPAN

044-855-1111, ext.3662

SPACE REQUIREMENTS FOR COMPUTER-ASSISTED INSTRUCTION

Frederick G. Knirk, University of Southern California

Daniel Christinaz, University of Southern California

ABSTRACT

Student learning environments involving computers require a large social investment. The better the "electronic classroom" environment is designed, the greater are the chances that learning will occur. The design/selection of acoustical, lighting, HVAC factors and space/furniture affect each individual. Each of these variables must be considered when designing solutions for instructional problems involving computers.

INTRODUCTION

The physical environment can affect student learning, affect and physiology. The design of learning spaces for cognitive, affective and psychomotor objectives is an instructional technology. Behavioral science data regarding color, form, acoustics, light intensity, light contrast, temperature can be used in ways which predictably affect learning via computer.

Educators and trainers need to know why to recommend specific facilities and must communicate these needs or specifications to architects and school authorities. Form should follow function! Too often a school is built by an architect interested in designing a beautiful building instead of building an environment for a specific instructional curriculum. In this paper we examine many of the environmental factors which facilitate or inhibit student progress.

THE ACOUSTICAL ENVIRONMENT

Tapping on the keyboard, clattering printers, and the whirl of cooling fans, as well as overhead conversations, and the continual shuffle of people walking by contribute to a distracting and disruptive environment. Many CAI tasks require intense concentration and creative thinking. They may require a workplace in which ambient noise is reduce, if not eliminated.

Noise levels influence an individuals ability to learn and be healthy. Generally hearing loss in excess of 70 decibels (the din of ordinary expressway traffic) will cause, according to the U.S. Environmental Protection Agency, temporary stress reactions to include increased heart rate, blood pressure and blood cholesterol levels. Lower intensity background noise may simply be annoying or result in distraction from desired learning tasks. Lower intensity noises, however, are not the detractors educators once thought them to be. In addition, short-term noise levels of 135 decibels may cause physiological effects such as nausea, fatigue, and a loss of muscular coordination.

Confusion exists about which of the several noise scales (A, B or C weightings) is best related to learning achievement. Noise is measured on three scales: A Scale (dBA), a scale that approximate the response characteristic of the human ear (this is the sound level scale used in most of the studies in this article and is to be assumed if not specifically stated); B Scale (dBB), a scale that intended to represent how people might respond to sound of moderate intensity; C Scale (dBC), a scale that weights all frequencies almost equally; Quiet condition. Lukas (1981) argues for the dBC scale. Classrooms are often found near roads and thus the background noise levels are frequently high, he argues. "Traffic, particularly that containing trucks, has a spectrum showing maximum acoustical energy at frequencies below about 150 Hz. These frequencies contribute much to C-weighted noise levels but are de-emphasized in A-weighted measurements." (Lukas, 1981, 14) ". . .it is quite clear that for both third and sixth graders the C-weighted descriptors of noise level are more highly coorelated with reading achievement than are the A-weighted levels." (Lukas, 1981, 29) The State of California suggests that a significant detrimental effect on the educational process begins when a noise level of 50 dBA is obtained, while the Federal Guidelines suggest such an effect begins at 63dBA. No such guidelines seem to be available for the C scale.

To provide some bench-marks on existing noise levels, the authors research provided the following readings for a typical elementary school: office (carpeted, A/C on, no student noises) 35dBA, 37dBB, 58dBC (aircraft and cars had the most impact on the C scale); in neighborhood (quiet, no cars, airplanes nor voices in immediate area) one block from school 58 dBC; noon recess outdoors near school, 75 to 80 dBC; independent reading (tiled classroom floor, 16 students, 1 teacher, 1 aide) 50 to 69 dBC with an average of 55 dBC; seatwork during math lesson 45 to 60 dBA with average of 50 dBA and 55 to 65 dBC with average of 60 dBC.

It is not only the absolute value of noise intensity that results in task avoidance and makes learning difficult, but the total range of intensity values (Corso, 1982), predictability of the occurrence of the sound and the relationship of the frequency of sound being attended to (teacher's voice) and the frequency of the other noise source(s) which result in distraction or learning interference.

"By choosing the right studies, one can show that noise produces either a decrement, no effect, or an improvement in performance." (McCormick, 1970, 440) "However, we can make a few generalizations about the effects of noise on performance. With the possible exception of some memory tasks, the level of noise required to obtain reliable performance effects is quite high, generally over 95 dBA. Performance of simple, routine tasks may show no effect and often will even show an improvement as a result of noise. The detrimental effects of noise are usually associated with difficult tasks which require high levels of perceptual or information processing capacity, or both." (McCormick, 1970, 440).

The detrimental effects of 1) increased confidence in judgment (certain decisions become more confident in the presence of noise), 2) funneling (person typically will focus attention on the most important aspects of a task or the most probable sources of information in a high noise environment, and 3) gaps (no opportunity for relaxation which will allow moments for low performance and gaps in performance) (Broadbent, 1976) have typically been associated with high levels of noise. "An apparent exception is performance on cognitive tasks. Here we often find a disruptive effect of noise at levels of 70 or 80 dBA. This is especially true of tasks involving short-term memory requirements." Weinstein (1977) for example, reported that 68 to 70 dBA noise signficantly impaired the detection of grammatical errors in a proofreading task (which requires short-term memory), but did not adversely affect the detection of spelling errors." Although there is beginning to be some consensus on the effects of noise, there still remains considerable controversy over why, or how, noise has the effects it does." (McCormick, 1970).

Salame and Wittersheim (1978) used four conditions (quiet, noise uninterrupted during the presentation of each list, noise emitted simultaneously with each list element, and noise emitted in the interval between each element) and a task of remembering a six digit list. At 96 dBA all noise conditions yielded more errors and omissions than the quiet condition, with the highest error rates occurring where noise was superimposed over the digits.

In a study by Wilding, Mohindra and Breen-Lewis (1982) involving 120 undergraduate students were shown a list of either nonassociated words or a list on which words could be organized into groups. The students were either given no orienting instructions or they were asked to rate the words for pleasantness (semantic-orienting task). In a 3-way interaction of noise by list type by orienting task it was found that 85-db white noise compared with 65-db improved performance on nonassociated list with no orienting task and increased recall in the original sequence, but had the reverse effect w/semantic orienting. With students who had the associated list, noise had no marked effects in nonorienting condition, but improved performance in the orienting condition. It was concluded that in noise maintenance, rehearsal tends to be adopted unless instructions induce an alternative strategy, and in the latter noise reinforces use of an alternative strategy.

A study by Pelletier (1984) studied the effects of continuous and intermittent pure tone noise at fairly low levels, 70 dBA, involving a task of a high cognitive level ("evaluation" in terms of Blooms's taxonomy). This study indicated that the noise conditions were significantly more detrimental on the score performance than the quiet condition. An unusual finding in this study was that there was no significant difference in the continuous or intermittent sound conditions. He found that high frequencies (8000 Hz) were significantly more detrimental than low freqquencies (500 Hz). No significant differences were found between groups when the quantity of work was examined.

Poulton (1978, 1068) contends that all the known effects of noise on performance can be explained by four determinates which combine to affect performance: 1) masking of acoustic task-related cues and inner speech; 2) distraction; 3) a beneficial increase in arousal when noise is first introduced, which gradually lessens and falls below normal when the noise is first switched off; and 4) positive and negative transfer from performance in noise to performance in quiet. Positive transfer results from the better learning of the task in noise under the influence of the increase in arousal. Negative transfer results from the techniques of performance used in noise to counteract the masking or distraction when they are not appropriate in quite.

Wheale (1982) exposed 20 19-39 year old males to 4 noise conditions: teletype, intermittent, jet-cockpit, and helicopter-cockpit noise at approximately 100dbA. The quiet condition was white noise at 66dbA. No decrease in task performance (4-choice psychomotor task) was found. The intermittent noise condition resulted in a significantly higher total error score when compared to other noise conditions. No significant increase in arousal was found, but as heart rate increased (a measure of arousal) misses decreased. KOR had a significant effect on arousal but not on performance. Extroverts scored significantly more errors than introverts. The main factors that determine degradation of performance under noise are 1) whether the task involved short-term memory rehearsal processes and 2) the fact that arousal may increase when students perceive that noise is interfering with the completion of an important task.

The State of California, in a federally funded program, studied fifteen elementary schools in Los Angeles to determine what impact freeway noise had on third and sixth graders learning ability in reading and mathematics. They concluded that the design criterion for traffic noise levels inside classrooms should be 58 dBC. The community noise level should not exceed 65 dBC. They found that "noise had a more systematic effect upon sixth graders than upon third graders and that noise appears to have more predictable effects upon the skills involved in reading than those in mathematics." (Lukas, 1981, 29) On an average, third graders in noisy schools were about 0.4 year behind in Reading, while the sixth graders were about 0.7 year behind. The largest effects of noise were observed thus: "at a constant noise level in the community, achievement in reading decreases directly with increasing noise level in the classroom." (Lukas, 1981, 32)

von Wright (1979) studied the effects on 36 six year olds and 36 nine year olds and 36 undergraduates who were asked to perform speeded card-sorting tasks. There was no interaction between salience and the effects of noise. Noise slowed the sorting performance of adults in all conditions; it slightly improved the performance of nine year olds and it improved the performance of six year olds significantly when irrelevant information was present. In a later study, von Wright (1980) asked 70 high school students to recall the names of countries during a five or eight minute period either in silence or in intermittent white noise (95 dbA). Noise interfered consistently with neurotic students but had little effect on stable students. Performance was not significantly related to extroversion nor to self-rated activation. Waters (1983) suggests that sound louder than 60 dB will interfere with concentration and a sense of well-being. There is little agreement as to that point where background noise or sound interferes with learning or concentration.

One way to control unwanted noise is through the use of carpeting. Sound can be absorbed by soft materials, nonflammable soft surfaces should be systematically installed. Nearly all acoustical surfaces are soft and have interconnected mazelike holes. Sound enters the holes and cannot get out. The sound energy is converted to heat as the sound molecules stop vibrating. Ceilings and walls may have to be treated, especially if interior spaces are small. Sound bouncing from hard surface walls can be as disruptive as that from the ceiling. The most important treatment area is from 2 to 6 feet from the floor. That is where most of the noise is produced.

Another factor affecting acoustics is layout. The shape of the individual workstation and the direction which it faces influences the direction that sound flows. Sound travels in straight lines and in cone-shaped fashion making the line of sight a good indicator of at least one path it will follow. Seeing lip movement, expressions and gestures of a person across the room increases intelligibility. Workstations should not be directly across from each other, but staggered to avoid direct sound paths. Panel components can be arranged so that speech and other sounds is directed into them. Techniques to ensure acoustical privacy also include enclosing or wrapping noisy equipment. Printers are often placed in soundproofing boxes (Harris, Plamer, Lewis, Munson, Meckler, and Gerdes, 1981; Cohen and Cohen, 1983; Pulgram and Lewis, 1984).

We highly recommend Wheale and O'Shea (1982) as a easy to read and logical summary of the impact of noise on performance. The booklet was written by the Swedish and translated to English by the Department of Labor. Many useful techniques for modifying noisy work environments are suggested.

In summary, reduce sudden, sharp, loud sounds such as mechanical noises from cars or trucks and animal noises and playground noises. The scraping of chairs on a hard floor in the CAI environment are a common source of classroom distraction. Background music noise may be used to mask these sounds.

LIGHTING

Lighting is one of the most important aspects of the electronic classroom. Too much or too little affects learning. Developing a good lighting system for the electronic classroom must ensure optimal luminous conditions for productivity at the required tasks and equipment, especially the use of VDT's. Repeated surveys and evaluation of health risks associated with VDT's use have shown that visual fatigue, eye strain, and burning eyes can be attributed in part to inappropriate lighting design. In general, research indicates that insufficient illumination, glare, reflectance, shadows, low brightness contrast, and flickering affect human performance and can be controlled by good lighting design. The environment should be visually interesting and support the comfort, health and safety needs of the worker.

Light levels should be related to the viewing difficulty of the task. While 30 to 75 foot-candles (fc) is required of a conventional classroom or study carrel to permit reading and desk-work, VDT screens should be in environments with 30 to 80fc depending on the readability of the associated materials and surrounding area. Adherence to raw foot-candle standards alone does not ensure either sufficient or efficient task illumination. Quantitatively you may have sufficient light by which to accomplish a task but conditions of glare, contrast loss, disturbing reflection, poor color rendition, and a generally "dull day" appearance can produce an environment that does not function properly. In the electronic classroom, especially where reflection from the VDT, is the primary source of visual discomfort to the worker, quality consideration are critical to lighting design; generally ambient illumination requirements are lower. Tasks at the work surface, however, will require increased lighting levels. If these levels are too bright it become difficult for the eye to accommodate the continual changes that occur while referring to a document and the VDT (Pulgram and Stonis, 1984).

Contrast is directly related to total brightness. The brighter an instructional area the higher the possibility of having a higher brightness ratio. If a student is confronted with a high contrast level, the pupils of the eye must continually attempt to adapt to variation in light level. The learning environment should be designed so that the student, when looking up from his or her work should be confronted with no more than 10 times the brightness of his work. A student who is attempting to read in moving shadows of leaves on a tree, or who is attempting to read white pages on a dark desk top will probably soon complain about a headache or "tired eyes."

Ceiling grids are the simplest way to ensure that the facility has sufficient light. Ceiling fixtures tend to disseminate even light. Hot spots are rarely created, and the color of the light isn't as dependent upon the color of the ceiling and the upper walls. Furthermore, some light reflects upon the color of the ceiling and that helps to evenly illuminate the learning environment. Ceiling systems may be very flexible. If the fixtures are on tracks or can be connected and disconnected at will they may be placed where needed (Cohen and Cohen, 1983). Room lights should be located more than 50 degrees above eye level so as not to be distracting. For more specific light levels for various types of learning activities see Knirk (1979, 91), or the GE Lighting Level Guide (#205-51333, 1986).

Glare is an extremely difficult problem for many video display users. There are two types of glare: direct and indirect. Direct glare is described as a light shining directly in the eyes. Usually the situation is so annoying, that it is fixed as soon as it occurs. Indirect glare encompasses light that shines first on one surface and then reflect from that surface into the eyes. Unfortunately, people can be bothered by indirect glare without being conscious of it. Those working at VDT's may suffer headaches, nausea, and even burning eyes from reflection on the screen. Indirect glare can be alleviated by the moving what is being looked at, changing body position, or task lighting. Task lighting systems tend to be very flexible; they are designed to bring light to areas only in which it is needed. The clip on fixtures can be moved around at will and can be switched on and off or dimmed with ease.

Understanding the impact or importance of color in the electronic classroom is difficult, and understanding what color is and how it is generated/perceived is even more difficult. ". . .colour depends not only on the stimulus wavelengths and intensities, but also on differences of intensity between regions, and whether the patterns are accepted as representing objects." (Gregory, 1978, 127) The perception of color involves high-level processes. Brown, for example, ". . .requires contrast, pattern, and preferably interpretation of areas of light as surfaces of objects (such as wood) before it is seen, and yet in normal life brown is one of the most common colours." (Gregory, 1978, 127)

Color is light, pigment, information, a form of energy, a range of electromagnetic radiations. The specific colors the eye sees depends on the wavelengths of light reaching the eye. Until it reaches the brain color does not exist; how the brain decodes and reconstructs in our minds a given color picture is yet inadequately understood.

Color affects learning. Color affects our ability to differentiate between objects on the VDT. Color preferences seem to affect learner's attention span. It seems possible to determine a student's color preference and then use in his/her learning environment (paper, desk top) and lengthen attention span (Phillips, 1965). Color is, however, not essential for learning; a great many successful individuals (8 to 10% of men) are color-blind. PLATO and many other CAI systems were very effective monochrome systems.

Intense hues (e.g., dark reds and blues) tend to cause restlessness and eyestrain when applied to a large area like an electronic classroom. Soothing neutral colors are most suitable. Some shades of green are considered to be calming and thus may be desirable for VDTs and electronic classroom walls.

In summary, the selection of paint for classroom walls or the color on VDTs influence learner attitudes and ability. If arousal or activity or discussion is desired select bright hues in active colors such as red, yellow, orange or rust. For calming students or getting them to reflect or integrate information use dull shades or quieting colors as blues, greens, gray or beige. In rest areas or dining areas or auditoriums use pink, peach or turquoise colors.

HEATING, VENTILATION AND AIR CONDITIONING

There is considerable evidence to indicate that temperature, relative humidity, air movement, and overall air quality have an effect on learning. "The effects of heat stress on performance in mental activities are entwined with related factors such as the type of task, the duration of the task, the degree of accommodation of the individuals, and the level of their training." (McCormick, 1970, 492) "When the socioeconomic status variable was statistically controlled pupil achievement in air-conditioned school buildings was consistently higher than pupil achievement in non-air-conditioned school buildings. Furthermore, in the vocabulary scores of ITBS, pupil achievement in air-conditioned school buildings was significantly higher than pupil achievement in non-air-conditioned school buildings at .05 level" (Chan, 1980). A study by McCardle (1966) also indicated that pupils in an improved HVAC environment made significantly fewer errors and required less time to complete the tasks than did pupils in the regular classroom. Harner (1973) found that with students between the ages of eight and eleven there is a significant decrease in reading speed and comprehension with classroom temperatures rise above 77° F.

SPACE AND FURNISHINGS

Despite the general preferences by teachers for classrooms with windows, research supports neither the claim that windowless classrooms will allow fewer outdoor distractions, and thus higher achievement, nor the fear that the absence of windows will have harmful psychological or physical effects. In one of the best studies in this area, Larson (1966) conducted a three year study of kindergarten through third grade students and found no significant difference in student attitudes nor learning when they were placrf in windowless environments. Electronic classroom designers may wish to consider the ability to control temperature, light level, and vandalism in windowless spaces.

Space requirements are the square footage necessary for the student, furniture, equipment, and ancillary space as well as allocations for circulation and the layout factor. Space is determined according to existing and projected requirements. These include (Pulgram and Stonis, 1984):

1 -- Tasks, that is, writing, referencing, thinking, filing, etc., usage.
2 -- Equipment, that is, VDT, microcomputer, printer, modem, etc.
3 -- Workstation requirements, that

is, full enclosure, semi-enclosure, or open.

4 -- Access requirement, that is, need for learner to learner interaction

5 -- Human factors, that is, ease of use, flexibility of components, adaptability to change.

An electronic workstation is a microworld and requires 10 to 20 percent more space than lecture spaces. Incorporation of a VDT or microcomputer requires a greater work surface depth than a conventional desk; there needs to be space for conventional tasks such as reading and writing as well as for referencing (printout sheets are often 11 by 14 inches) and other conventional electronic equipment; and there are special requirements of electronic equipment, such as extra circuits, storage for electronic media, and ventilation.

Additionally there is the need for personalization and personal comfort. Self expression allows the student to "personalize" the workstation by being able to control intrusion and establish "ownership" rights. The "bullpen" concept totally exposes the student to all manner of visual, acoustical and physical interruption. Visual control prevents the student from being seen, minimizes physical interruptions and contributes to a feeling of having personal space. Eye contact is tantamount to conversation, or at least recognition of another human being and can be disturbing (Cohen and Cohen, 1983). Individuals do not like to be approached to from behind, it is especially unpleasant if the work-station orients the occupant to the walkway. Acoustical control lessens unwanted and distracting machine or conversational noise to and/or from the workstation. However, remember that in planning for personal comfort to avoid achieving visual and acoustical control at the expense of visual interest and spatial clarity. Avoid creating an environment of boredom (Pulgram and Stonis, 1984).

Conventional furniture -- desk, chair, bookshelf, and file cabinet -- is a collection of furniture, while a workstation is an integrated system of interdependent furnishings. It is a personalized mini-environment within the larger classroom. The basic furnishing include:

1 -- Chair(s)
2 -- Worksurface
3 -- Storage--shelves and/or cabinets
4 -- Electronic Equipment--VDT, microcomputer, etc.
5 -- Privacy/acoustical screening

The primary objective in selecting the correct chair for a workstation is personal comfort. The correct chair needs to be adjustable, that is, the chair must fit the individual in a variety of position and for a variety of task. The ability to change position is essential. It is difficult to for anyone to sit still in one place for any length of time. Most chairs allow of for back attitude control which locks the chair into several different positions. The appropriate position balances the user's weight between the upper torso, supported by the chair back and the feet. Almost all chairs feature height control; the seat can be moved between 13 and 20 inches above the floor. The proper height allows the feet to rest flat on the floor while enabling the thighs and forearms to remain horizontal. Another ingredient in picking the correct chair is seat depth and width. Seat depths should allow for adequate support of the thighs while adequate width stabalizes the trunk and evenly distributes body weight. Proper seating allows the body weight to be supported primarily by the bony structures of the buttocks while reducing stress on the back and leg muscles. Prolonged, uneven pressure will eventually lead to back problems (Salmon, 1979; Cohen and Cohen, 1983; Pulgram and Stonis, 1984).

The workstation must support the comfortable use of computer related equipment and the efficient performance of total task requirements. A common complaint is that there is little workspace to do paper and/or pencil tasks. Electronic equipment does not do away with paper and/or pencil tasks. The work surface widths must support the primary or most frequently used equipment, any other equipment must be readily accessible and all immediate support functions. The work surface depth is determined by the total depth dimension of the largest piece of equipment and lateral rear clearence for equipment access and ventilation or by a specific task requirement. The work surface height should be adjustable so as to reduce the chance of musculoskeletal fatigue and provide for a correct VDT viewing angle. The VDT should be positioned at an angle of 20 to 30 degrees below the horizontal eye level. The top of the screen should be 10 degrees below horizontal. The wrist should not incline at an angle in excess of 10 degrees above or below the keyboard, while the forearm is horizontal to the home key row (Pulgram and Stonis, 1984; Salmon, 1979).

Carpeting is a versatile, flexible material available in a wide range of specifications. It upgrades the acoustical environment and provide lower maintenance cost than other floor coverings. Carpeting upgrades the acoustical environment by stemming noises radiating from computer equipment and voices, from objects scraping surfaces, and from impact from foot-falls and/or objects striking the floor.

Certain carpeting material have high static electricity potential when they are in contact with one another. PVC, oak tanned leather, and neolite are common material used in the manufacture of shoe soles. Wool and nylon have a high static electricity potential when in contact with these materials. Carpeting must be static resistant. Static electricity is reduced when a conductive path from the body to the ground is provided for the electrical charge. The most common method is to make carpeting electrostatically conductive is to include a carbon infused antistatic fiber for ever four or five nylon fibers (Cohen & Cohen, 1983).

An article in The Harvard Medical School Health Letter (Johnson, 1983) suggested that the health problems associated with VDTs fall into four categories: Alleged radiation hazards, visual disturbances, musculoskeletal difficulties, and job stress. This section dealing with human factors considerations in using computers in education will generally follow this outline: 1) radiation hazards, 2) visual aspects of reading computer screens and printout, 3) musculoskeletal considerations, and 4) learner stress as it relates to using computers in schools.

There is no **radiation hazard** from VDT's (Johnson, 1983; Rowinsky, 1987b). The impact of viewing VDT displays has come under scrutiny by researchers in many nations. As the number of VDTs has increased in industry, the number of worker complaints of eye strain and eye fatigue, blurred vision, neck and back aches and a number of psychosomatic disorders are often reported. In 1981 the National Academy of Sciences in Washington examined the evidence supporting claims of cataracts and other potentially related health problems, and said that the amount of ionizing and non-ionizing radiation emitted by VDTs is too small to pose a threat of either cataracts or birth defects.

With regard to **visual disturbances** Johnson (1983) concluded that people working with VDTs "...report discomford or difficulty with their eyes more often than other workers with visually demanding jobs." (p. 1) The letter suggests five minimal criteria for the VDT: a) the image should not flicker, b) the entire display should be in sharp focus (no blurred edges), c) the contrast between light and dark areas should be at least 8 to 1, d) the characters should be formed by a 5x7 dot matrix at the very least (5x9 is becoming the standard), and e) brightness and contrast controls should be easily adjustable by the VDT user. (p.2)

A study by Muter and others (1982) involving thirty-two subjects who were asked to read either from a VDT or from a book for 2 hours experienced little dizziness, nausea or headache in either condition. VDT subjects read 28.5% slower than the book subjects. The authors suggest three reasons for these slower speeds: 1) The subjects were familiar with books and reading from books but not from VDTs, 2) The books had approximately 400 words per page and the mean number of words on the video page was approximately 120; subjects may have a tendency to consume pages at a fixed rate, 3) The number of characters per line differed; the VDT had a maximum of 39 characters and the book condition had approximately 60 per line. Kolers, Duchnicky and Fergusen (1981) found that reading speed on a VDT was 17% faster with 80 characters per line than with 40 larger characters per line. Proportional spacing had no effect on reading speed or comprehension. This study suggests that extended reading from television screens or computer VDTs is certainly feasible. Mourant, Lakshmanan and Chantadisai (1981) found that VDT usage for 2 to 3 hours had a measurable fatigue impact on the learners. Fatigue, however, was found to be readily reversible when breaks were provided every hour. Kearsley and Hillelsohn (1982) conclude evidence suggests that if an individual wants to spend many hours watching a VDT screen, this is possible without any real performance decrement although individuals will likely experience visual fatigue after a few hours without a break. "Thus, session length is determined by the individual and by the designer in terms of content considerations. Typically a lesson designer will be able to estimate the length of a lesson or course on the basis of the number of displays allowing time for reading, calculation, thinking, etc." Kearsley and Hillelsohn, 1982, 77)

Curriculum designers/developers must take these computer-related potential problem factors into consideration when designing CAI activities and provide for a variety of activities to supplement computer related activity. Health problems caused by VDTs can probably be averted by planning that takes account of the student's needs. Lighting and furniture at the worksite may need to be rearranged or redesigned to prevent eyestrain and postural stress. The learner should have frequent breaks so they can move about.

SUMMARY

The physical environment within which we work as professional educators will affect our effectiveness. So also, will the learners' environment affect how well

well he can learn. Since we can use temperature, color and acoustical variables to affect the learner on a predictable basis, we really have qn instructional technology. Instructional technologists must know how to design environments and how to work with architects and the school plant personnel so that new or remodeled facilities are designed to match the objectives, learning strategies and media requirements.

A VDT operator will generally find that it is easier to dim the lights to maximize the contrast between the image and screen and to reduce contrast between the overall VDT light level and the wall or table reflectance.

To direct learner attention to specific areas on a VDT screeen it is desirable to use colors or boldfacing. This technique is useful, for example, in directing students attention to the options in a menu. It is not recommended to use blinking or flashing information (Heines, 1983, 75). These last techniques are so strong that they make it hard to read the other material on the screen.

Heat is also generated by a VDT terminal. The terminal produces as much heat as an individual (about the same as the heat produced by a 100 watt incandescent bulb). Additional cooling is often needed to handle this increased amount of heat.

The sound level is an office using computers should also be considered. While VDT terminals are less noisy than a conventional typewriter, the impact printers used to provide hardcopy may generate 70 decibels' worth of sound -- twice the level considered comfortable in an office setting (Waters, 1983, 74). This problem may be reduced by either locating the printer away from the work area or by installing sound-absorbing materials near the printer.

Environmental considerations affect the design of almost all instructional technologists' activities; managing a program, designing an instructional set of materials, evaluating an instructional situation or whatever else he or she may do.

REFERENCES

Broadbent, D. (1976). Noise and the Details of Experiments: A Reply to Poulton. Applied Ergonomics, 7, 231-235.

Chan, T.C. (1980). Physical Environment and Middle Grade Achievement. Greenville, SC: Greenville County School District.

Cohen, E., & Cohen, A. (1983). Planning the Electronic Office, McGram Hill Book Co., New York.

General Electric. (1986). GE Lighting Level Guide, #205-51333. Cleveland, OH: General Electric.

Gregory, R.L. (1978) Eye and Brain. (Third Ed.) New York: McGraw-Hill.

Harner, D.P. (1973). A Review of Research Concerning the Thermal Environment and its Effects on Learning. Dissertation, University of Mississippi.

Harris, D.A., Plamer, A.E., Lewis, M.S., Munson, D.L., Meckler, G., & Gerdes, R. (1981). Planning and Designing the Office Environment. New York: Van Nostrand Reinhold.

Heines, F.M. (1983). Screen Design Strategies for Computer-Assisted Instruction. Maynard, Mass.: Digital Equipment Corporation.

Human Factors of Workstations with Visual Displays. (1984). (3rd ed.) Armonk, N.Y.: International Business Machines Corporation.

Hirsch, R.S. (June 1984). VDTs and the human factors community: Tipping the Iceberg, Human Factors Society Bulletin, 276, 1-3.

Johnson, G.T. (ed.) (April 1983). VDTs - a new social disease? The Harvard Medical School Health Letter, 8(5), 1-5.

Kearsley, G.P. & Hillelsohn, M.J. (May 1982). Human factors considerations for computer-based training, Journal of Computer-Based Instruction, 8(4), 74-84.

Knirk, F.G. (1979). Designing Productive Learning Environments. Englewood Cliffs, N.J.: Educational Technology Publications.

Kolers, P.A., Duchnicky, R.L., & Ferguson, D.C. (1981). "Eye Movement Measurement of Readability of CRT Displays." Human Factors, 23, 517-527.

Larson, C.T. (1966) The Effect of Windowless Classrooms on Elementary Children. Ann Arbor, MI: Architectural Research Laboratory, University of Michigan.

Lu, C. (January 1983). Will there ever be an efficient keyboard? High Technology, 3(1), 22-23.

Lukas, J.S., DuPree, R.B., & Swing, J.W. (September 1981). Effects of Noise on Academic Achievement and Classroom Behavior. Sacramento, CA: California Department of Transporation.

McCormick, E.J. (1970). Human Factors Engineering (Third Edition). N.Y.: McGraw-Hill Book Co.

McCormick, E.J. (1976). Human Factors in Engineering and Design (Fourth Edition). New York: McGraw-Hill Book Co.

McCormick, E.J. & Sanders, M.S. (1982). Human Factors in Engineering and Design (Fifth Edition) N.Y.: McGraw-Hill Book Co.

Mourant, R.R., Lakshmanan, R., &

Chantadisai, R. (1981). Visual fatigue and cathode ray tube display terminals, Human Factors, 23(5), 529-540.

Muter, P., Latremouille, S.A., Treurniet, & Beam, P. (October 1982). Extended Reading of Continuous Text on Television Screens. Human Factors. 24(5), 501-508.

Norman, D.A. & Fisher, D. (October 1982). Why Alphabetic Keyboards Are Not Easy to Use: Keyboard Layout Doesn't Much Matter. Human Factors. 24(5), 509-519.

Pelletier, P.N. (1984). Comparison of the effects of continuous and intermittent pure tone at low and high frequencies on reading comprehension. Unpublished doctoral dissertation, University of Southern California, Los Angeles.

Phillips, M.M. (1965). The Psychological Effects of Color on the Attention Span and Reading Ability of the Hyperactive Child. Master's degree project, University of Southern California.

Poulton, E. (1978). A New Look at the Effects of Noise: A Rejoinder. Psychological Bulletin, 85, 1068-1079.

Pulgram, W.L., & Stonis, R.E. (1984). Designing the Automated Office: A Guide for Architects, Interior Designers, Space Planners and Facility Managers, Watson-Guptill, New York.

Rowinsky, W. (1987a). Quality of monitor affects productivity, comfort of workers. PC Week, 4(1), 99, 102.

Rowinsky, W. (1987b). Advocates and scientists disagree about hazards of CRT-based displays. PC Week, 4(1), 99, 106.

Rudin, M. (October 1982). The Ergonomics of Computer Terminals. Computer Products. 64.

Salame, P., & Wittersheim, G. (November 1978). Selective Noise Disturbance of the Information Input in Short-Term Memory. Quarterly Journal of Experimental Psychology. 30(4), 693-704.

Sinofsky, E.R., & Knirk, F.G. (1981, March). Color; It can make a difference. Instructional Innovator. 26(3), 17-19.

Salmon, G. (1979). The Working Office. London: Design Council Publications.

Tarter, J. (June 1983). Ergonomics: a primer, Business Computer Systems, 55-72.

Turner, J.A. & Heller.R.S. (November-December 1982). Selecting the Proper Compter Work Station. Today. 27.

von Wright, J., & Nurmi, L. (March 1979). Effects of White Noise and Irrelevant Information on Speeded Classification: A Developmental Study. Acta Psychologica. 43(2), 157-66.

Waters, C.R. (January 1983). Just When you Thought it was Safe to go Back to the Office. Inc., 69-74.

Wheale, J.L. & O'Shea, N.M. (November 1982). Noise and the performance of a four-choice psychomotor task, Ergonomics, 25 (11), 1053-1964.

Wilding, J., Mohindra, N., & Breen-Lewis, K. (November 1982). Noise effects in free recall with different orienting tasks. British Journal of Psychology, 73(4), 479-486

F.G. Knirk
USC School of Education
EdPT WPH 801
Los Algeles, CA 90089-0031
(213) 743-6263

A NATURALISTIC FRAMEWORK FOR THE DEVELOPMENT OF A COMPUTER ETHICS

DR. DONALD SCHERER

BOWLING GREEN STATE UNIVERSITY

ABSTRACT

The thesis of this paper is that by examining the environments computer use is creating, we can discover the norms appropriate for achieving social coordination. The paper sketches a conception of ethics as a part of the evolutionary development of human culture. The (social and natural) environments in which people live are presented as interdependent with the norms appropriate for achieving social coordination in those environments. It is argued that norms have developed for dealing with what is absent, with what is future and with the self-concepts which grow out of self-reflection. The usefulness of those norms for computer environments is suggested.

For the first time in its history, the twenty-eighth annual convention of the association for the Development of Computer-based Instructional Systems included a forum on ethics. Both the panelists and those recognized from the floor suggested many important normative questions surrounding the use of computers. If educators need time to preview programs and producers need to protect their markets from copying, how should the educators and the producers treat each other? When has one properly acknowledged the work of others from whom one's own computer program benefits? And will the solitary nature of some computer uses warp the moral development of young users?

The difficulty many people felt with the session was that, diverse as the questions were, the session provided no hint of a framework for relating the questions. How might one relate questions about the rights of developers of computer programs to concerns about the effects of computer use on children's socialization? The proposal of this paper is that an illuminating naturalistic framework for understanding ethics can be developed out of contemporary work in anthropology, archeology and philosophy, an understanding which has both the advantage of fitting within a broadly scientific conception of the world, while at the same time relating the diverse humanistic concerns implicit in typical moralities. My tactic in this paper will be to describe what I take to be the historical advent of the elements of morality. I will use the example of the educators and the producers to illustrate how these same historical elements arise in contemporary problems concerning computers. And since I will describe the historical processes of responding to practical problems in what we would describe as moral ways, I will show much of the range of tactics available for responding to contemporary problems concerning computers.

The widespread introduction of computers into the education and work of people means the creation of new environments within which people function and interrelate with one another. From the solitary wordprocessor to interactive computer-assisted instruction to the interaction of telecommunications, computer uses create new learning and working environments. The problem about new environments is that any moral norms are designed for working in particular environments which afford some reinforcement system for them. I, for instance, live in a small town where the local newspaper daily prints "The Blotter," a column detailing local arrests. Because the town is small, enough people know each other that the publication serves a deterrent function. How large can a town become before this reinforcement system becomes ineffective? The question we need to address, regarding the new environments computer work creates, is when they should be controlled so as to maintain the viability of old norms and when the norms should be remolded because of new environments. If, e.g., the traditional right to control the use of one's property (say, one's computer program) is undermined (by copying technologies) should one redefine what ownership is or should we attempt to prevent copying so as to preserve our traditional idea of ownership?

To answer such questions, I choose to sketch some of the naturalistic framework for understanding the development of norms within human life. It is apparent that hominids have lived a social existence for at least two million years. (Leakey, 1977, 98) The million and a half year old remains of a hominid brain of 900 cc. have been found. (Leakey, 108) Over the next 1.4 million years, that brain size steadily grew, reaching 1600 cc. 100 thousand years ago (Brace, 1979). The implication of evolutionary theory is that such a rapid change would never have occurred except as the hominids were evolutionarily advantaged by increased brain size. Yet in the last 1000 thousand years, brain size has diminished from an average of 1600 cc. to the present average of approximately 1400 cc. (Brace, 1967, 108).

Why then has the evolutionary tendency to larger brain size not remained advantageous? If we start from the simplest of hypotheses about the role of the brain in the organization of behavior, it is plausible to hypothesize that increased brain size might well have sophisticated behavioral repertories which made environment-manipulating omnivores more dominating in original environments and successful in a larger range of environments. But such hypothesizing simply exacerbates the question of why brain size has actually diminished.

Here C. Loring Brace has argued from an analogy to computers. The size of a computer depends centrally on the efficiency introduced by its C.P.U., which functions to organize and process data. What then might have introduced such efficiency into the human brain? The only plausible hypothesis is that the evolution of language introduced the comparable efficiency into the operation of the human brain.

Like a C.P.U., the use of language will obviously have an important effect on the development and transmission of information. The primitive weapons and tools of human culture, which had remained stable for hundreds of thousands of years prior to 100,000 years ago, change thereafter in a marked and cumulative way. Similarly, the invention of pottery and the revolution it brought to human diet (and human dentition) dates to approximately 90,000 years ago. Certainly the rapid evolution of physical culture which began in the millennia in which the human brain began its decrease in size reinforces the hypothesis of the genesis of language with broad information-transmission capabilities. But since my focus in this paper is on ethics, I want to concentrate on the use of language to regulate behavior. For although it would be naive to maintain that morality is no more than the use of language for the regulation of behavior, it is my hypothesis that we can most plausibly model the development of ethics out of the regulation of behavior.

Animal behavior studies show that the regulation of behavior obviously has prelinguistic roots. If one concentrates on the social primates, one finds that the range of instinctual behaviors is much narrower there than in other mammals. C. Owen Lovejoy hypothesizes that the social behavior of hominids co-evolved with increasing maternal care, increasing intelligence, increasing brain development, decreasing number of offspring, extended childhood, increased play and increased group size. (Johanson, 1981, 327) By contrast, if one examines the social insects, one finds that the coordination which their societies require seems to be entirely instinctual, thus rendering regulative behaviors superfluous. In social primates, however, the behavior of mothers toward their young and the behavior of dominant males is highly regulative.

But what is the contrast between regulation possible with and without language? The contrast is in the salience which language makes possible. Imagine a woman being directed by someone speaking a foreign language with the result that she is left without any clue from her own language of what is being said. In such a situation the woman is forced to cast about (for what she takes to be the) features of her environment and for what she takes to be the import of the gestures of the speaker. Because she has a language of her own, she can even hypothesize that the speaker is "pointing" "to" the "building", not "in the direction of" the "sunset," etc. Being a language speaker, the woman may realize that a speaker may point, referring once to a thing, once in a spatial direction, once as apart of contrasting different categories of thought, and so on. But since the woman we have imagined does not know the language, she is initially left to guess which meaning the speaker conveys. That is, although without language one agent may wish to convey the acceptability or unacceptability of a particular behavior, the lack of linguistic clues leaves the hearer to intuit what about a particular behavior makes it (un)desirable.

Similarly, one non-linguistic animal may continue nudging another until the other's behavior changes in what we would describe as "some particular way," but the idea that the nudging animal wants the other to "stay close" is a human posit, as is the view that the nudged animal "realized" it was to stay close. In such cases we do not precisely anthropomorphize; rather we linguisticize the animal response: for the use of language creates a selective focus which adds a salience to the "appropriateness" of some description of some human behavior. This salience is lacking, no matter what behavioral criteria are met, when a non-linguistic critter's behavior is described. We have only behavioral reasons for saying that a dog responds to tone of voice, not to the meanings of the words.

How is the use of language a salience-creating device? When two persons speak the same language, the use of language directs the attention of language speakers to particular characteristics of their environment. A language user can say that we should not think of a "physical disk" as the "property" of a producer; we should think of the "computer program" as something the reproduction of which the producer has the "right to control." If social animals are going to coordinate with each other non-instinctively, then they must interpret the sensory stimuli they receive from their environments and from each other as indicating the appropriateness of a particular harmonious interactive pattern. The development of language, then, allows social animals to make various features of circumstances salient by naming them, thereby facilitating the learning of norms. (Is it not plausible that the veneration of the power of naming, which extends into human pre-history, has its roots in how early language users marvelled at the power of words to create salience?)

A related point arises from a practice anthropologists cannot date but which is probably almost as old as language: the use of something approximating definitions. For if there is a language, there is the possibility that a hearer will not understand a particular communication. And if there is the possibility of misunderstanding, the need arises for the misunderstanding not to prevent some desired coordination. Obviously, non-linguistic means of handling this need will be possible, but just as obviously a misunderstood animal will often repeat itself. And sometimes when the misunderstood animal repeats itself, it will repeat itself "in other words," as we say. The indifference of the speaking animal to the alternate locutions will then become an operational understanding of synonymy, and the beginning of a practice of defining will have emerged. If the use of other words in the face of misunderstandings proves efficacious, then the general power of language to call attention to features of the "environment" will be invoked to call attention to a particular linguistic aspect of the human environment: the use of other words to overcome misunderstandings. (It seems probable that this would be the first sense newly linguistic animals would gain of the recursive power of language.) The use of words in this way will then evolve toward the practice of seeking alternate linguistic expressions. When this practice becomes well-known, it may be made even more salient by being given a name, "defining."

The facility of language for coordinating human interactions is not, however, limited to coordinating the focus of human attention on previously existing features of the human environment or previously existing features of the human interaction. Consider what had to happen to language with the coming of the agricultural revolution. Worldwide we know that nomads do not own real property. But in the course of human evolution nomadic peoples have continually settled out. This apparently never happens so long as food supplies are ample to the travelling band, but when population increases to the point of making food supplies tight, nomadic peoples begin the rudiments of agricultural practices. (Ross, 1987)

Here a problem may arise: any agricultural process involves an investment of labor which bears fruit only at some later time. Thus in the intervening moments those who have not invested the labor may arrange to benefit from it. Obviously if such interventions are regular, the interventions will either discourage the development of the agriculture practices or encourage arrangements for preventing the interventions from being effective. If we take seriously the empirical premise that food scarcity drives the development of agriculture (Harris, 1970, 29), it is no surprise then that practices arise for guaranteeing that those who labor benefit from their labor. These practices range from boundary demarcation to security systems, even before the advent of written language. It is the protection of cultivated lands which leads headmen to create the specialized social role of protector, a role which is the precursor of both the police and the army. (Harris, 69f.)

In the creation of this specialized role, we encounter a remarkable development of human history, the delegation of authority. Undelegated authority among primitive peoples rests with persons whose attributes "fit them", as we say, for the specialized role they play and are accepted as playing. (Oliver, 1955, 399ff.) In contrast, delegated authority rests in part on the ability of the undelegated authority to choose wisely, where the ability of the delegated authority to play the delegated role is only one of the desiderata informing the wisdom of the choice. Because having the delegated role played at all may be more beneficial than not having it played, the idea of right (or acceptable) action grows in complexity with the creation of delegated authority.

The idea of real property, already complex, gains in complexity with the introduction of written language. For, obviously, a title or a deed to a piece of real property is a written record through which some authority certifies in a publicly accessible and publicly retrievable way an agreement that henceforth some individual shall have an authority over that land not shared by others.

To see more clearly the role of records in the regulation of behavior, let us consider how written language originated. The earliest extant written records are Phoenician shipping consignment list. (Others, defining written language somewhat differently, assert that mesolithic humans kept calendar records for seasonal purposes. If this accepted, the point I make below about absence remains.) When goods are placed on ships, ship captains frequently do not have anything of comparable worth which they can exchange for the goods placed on their ships. Clearly, exchange will be facilitated--even in a barter society--if ship captains can use goods they later receive in distant ports to compensate those who earlier put goods on their ships. But the temporal element in these arrangements creates the special problems of absence. For when the ship captain returns to port, nothing may remain in the environment of the port to show what goods were placed on his ship months earlier. Or, if anything remains, it may be only the memories of persons present both earlier and later. When expressions of memory conflict, however, a conflict about proper compensation will arise. Social order and the advantages of continuing commerce will then push toward making changes in the environment when the goods are put on the ship which changes will last until the ship has returned. In other words, the practice of record making will be born. For the practice of making a record is the practice of manipulating materials which, once manipulated, will retain their new, publicly accessible configuration as long as is practically desirable. One thought, then, about educational professionals who make copies of programs is that if disks sent out for preview were so marked that copying them created an observable disparity, then that observability would enforce the trustworthiness of the previewing professional.

The practical problem of consigning goods to ships, however, does not consist simply in the problem of the unreliability of memory about which goods were placed on which ships. For the goods were placed on the ships with the understanding of an appropriate quid pro quo. Consignment is, after all, a practice built upon more primitive exchange practices. Exchange, as a voluntary relationship, requires some sort of mutual satisfaction, for the motivation of voluntary exchange must be mutual preference of what the other has to what one has. But with its power to make obvious the need for creating a salience not present in an environment, language can be used to present the idea that what one party will have will be a satisfactory exchange for what the other party already has or will have. This thought is the root of contract. For the construction of projects (actions whose desired results are temporally removed) creates a problem more complex than we have allowed ourselves to discuss in the contexts of agriculture and consignment: the existence of projects creates the conceptual possibility of the exchange of that which is not present now. As we discussed concerning the advent of agricultural practices, the temporal gap creates the problem of the risk of intervention and the need for security, while the temporal absence of what is to be exchanged creates the problem of precision in defining what it is that is not present but is to be exchanged. Contracting allows for other models for resolving the preview problem: one might contract in advance to take some of the many programs one previews. With an appropriate pricing structure, the contract would guarantee an appropriate exchange for producers.

Notice how the elements of what we still take contracts to be respond to the problems such exchanges create. Because contracts are records of agreements, they create a salience in the later environment which encourages compliance. Because contracts have witnesses, the salience is harder to deny and the reliability of the record itself is vouchsafed. Because contracts have sanctions, there is also a response to the inevitability of risk associated with the future, independent of human best efforts. (And here is the genesis of the concept of compensation.) Because sanctions fall upon particular parties, incentives are created for minimizing the risks. (And here is the genesis of a concept of responsibility extended beyond the bounds of kinship and affection.) Because the mutual satisfaction goes unattained unless by some specifiable future time the party of the second part gains the satisfactions due, contracts have terms. The roots of the concept of contract, then, are totally contained within the material advantages and risks of exchanges of something now for something later. Contracting is a constitutive relation built on these material roots.

Another common use of language among primitive peoples is to recount stories. Whether these stories be historical, legendary or mythical, the stories are repeated by peoples, with the result that the role of the story teller is created and oral traditions develop. If one looks at the classic civilizations of Greece, Israel, India or China, one finds the development of oral traditions. In each of these traditions, the stories are eventually written and at least the more privileged males in these successful cultures have the leisure for reading these texts, reflecting upon them and memorizing much of them.

The process of reflection creates a new inward focal point for action. It is noteworthy that the most ancient of written codes are entirely behavioral in content. For instance, the original form of the Ten Commandments has, rather than the commandment "Thou shalt not steal," the commandment, "Thou shalt not steal a man." That is, the original commandment prohibited not theft but the kidnapping of free men. Correspondingly, the original meaning of "covet" was "steal," so that the commandment prohibited stealing one's neighbor's house, his wife, his manservant, his maid servant, his ox, his ass or anything else he might own. The form of the commandments which comes down to us in Exodus 21 generalizes the prohibition against stealing to cover both kidnapping and property, thereby freeing the commandment about "coveting" to be reinterpreted more inwardly.

What the art of story telling began and the process of reflecting on texts developed was the creation of the concept of vicarious experience. What would I have done in his position? What must it have felt like to have to make such a choice? And in the light of these questions, a different breed of thoughts arises: What kind of strength must he have had within him that he could have acted so well? (The word 'fortis' from which we derive 'fortitude' means 'strength' as I use the term here.) What is the ideal of manliness within him which allows him to act so well in those circumstances? (The word 'virtue' etymologically means 'manliness.') What traits of character distinguish the man we rightly idealize from other men? Would that through my meditation on these treasured stories I could become such a man myself! (Psalm 1 describes such a meditation man.)

This new breed of thoughts grow in classic literatures into a cultivated inwardness, into an identification of the superiority of our people over other peoples because of our civilization, whose root is our cultivated inwardness, and into the identification of persons in terms of the person's identification with a set of virtues, the latter identification itself grown out of cultivated inwardness. The common theme of Confucius, the Upanishads, the Israelite prophets and Greek playwrights is that the worth of a human life is measured by the kind of integrity it develops and maintains. This discovery of the soul mightily expands the domain of ethics by requiring that the justification of norms should include not only the effects of external behaviors but also the effects of the taking of the action on the self-consciousness of the agent. The concepts of dignity, self-respect and autonomy, which seem so central to the modern consciousness, take their original shape out of the vicarious experiences which arise from sustained reflection upon the stories of the people. Indeed, one of the common concerns surrounding computer use is the perception that so many computer users seem to have no moral compunctions about what they do. But, on the other hand, we know that people do have compunctions about how they are regarded within small peer groups. Accordingly, it has been proposed that school superintendents sign contracts with vendors for the trustworthiness of their teachers in program previewing. Part of the thought here is that the superintendent's personal relations with school principals and the principals' personal relations with school teachers will create environments in which conscience will be an effective forcer. Another part of the thought is that if school superintendents are both encouraging teachers to preview programs and discouraging teachers from copying programs, then they must themselves take some responsibility for providing reasonable funding through which teachers can purchase desired programs. And, finally, the thought is that if producers are going to be able to access the educational market, where multiple copies are often required and where funding is scarce, then pricing policies and licensing agreements will be needed to provide superintendents with appropriate leverage with their teachers.

Philosophical ethics concerns itself with the questions of what basic varieties of values exist and what the basic principles of right action are, given the varieties of values. We trace the origin of philosophical ethics to ancient Greece and to an era of cosmopolitan life in which variety within recognized values was compounded by various ideas of right action, all of which variety induced in certain itinerant rhetoricians the view that standards of right action are merely conventional and relative. Philosophical ethics has henceforth been fed by conflict within societies about principles of right action.

Problems of ethics, then, are problems about interactions completed only in the future, interactions often complicated by absence. They are problems which tend to presuppose human reliance on the salience which language creates for achieving coordinations of behavior, on the patterns which conventional acts like consigning, recording and contracting familiarize and regulate, and on internalization of value and the resultant development of personal moral integrity. In any ethical problem, I am proposing, the challenge is to use the salience of language, the patterns of conventions and the internalizations of value in such a way that these powers maintain and reinforce each other and promote mutually desired human interactions despite their complexities. I have tried to illustrate how this is the challenge in problems of computer ethics.

Cultural pluralism (when combined with human mobility) and technological innovation are two prominent forces fueling conflict within our own society. For the operation of both of these forces brings about dissonance between the circumstances in which some set of norms arose and the circumstances within which we live. This dissonance creates two tendencies, the very two tendencies we saw in operation when scarcity of food for nomadic peoples led to the beginnings of agriculture: on the other hand is the tendency to make the circumstances revert so as to take a form which eliminates conflict between the norm and the circumstances; on the ohter hand is the tendency to alter the norm so as to retain perceived value while accommodating changed conditions. The normatively strong reason for controlling the circumstances will be the value inevitably jeopardized by allowing the circumstances to vary. (If property rights cannot be respected, a producer may note, then exchange will be inhibited.) Often that value will be defined in terms of the impact of circumstances on the "kind of persons we are making of ourselves." (What kind of a world is it, a teacher may lament, when my budget tells me I must either steal from the creator of a worthy teaching tool or set aside my dreams to be the best teacher I can be?") The strong reason for altering the norm will be that, given changes in material circumstances, the maintaining of the norm jeopardizes the efficiency and effectiveness with which basic human needs are met. ("Without a separate pricing policy for the educational market, there will be no educational market.")

Consequently, the goal of conflict resolution in ethics is to avoid having to choose between the meeting of human needs and the maintenance of those social practices which have enhanced human identity and the richness of human society and human life. Maintaining supplies to meet demand (production incentives), controlling demanders to balance supplies (population control), substituting materials and procedures which do not endanger the quality of human life (pollution control, where some pollutants are material, some social) and redistributing opportunities and materials so that recognized basic goods are not denied to some because of the failures of the first three are the ideal forms of action, because they are actions which encourage not needing to choose between values. While much ethical discussion is given to how to handle the necessity of choice, the focus of the good life and of the good society is on the overcoming of this necessity.

FOOTNOTES

Brace, C. Loring, "Biological Parameters and Pleistocene Hominid Life-Ways," in Primate Ecology and Human Origins, I.S. Bernstein and E.O. Smith, eds., NY, Garland Press, 1979.

Brace, C. Loring, The Stages of Human Evolution, Englewood Cliffs, Prentice Hall, 1967.

Brace, C. Loring, unpublished lecture, Bowling Green State University, 1986.

Harris, Marvin, Cannibals and Kings, NY, Random House, 1970.

Johanson, Donald, Lucy, The Beginnings of HumanKind, NY, Simon and Schuster, 1981.

Oliver, Douglas, A Solomon Island Society: Kinship and Leadership Among the Siuai of Bougainville, Cambridge, Harvard University Press, 1955.

Ross, Eric, "An Overview of Trends in Dietary Variation from Hunter-Gatherer to Modern Capitalist Societies," in Harris, Marvin and Ross, Eric, Food and Evolution, Philadelphia, Temple University Press, 1987.

Donald Scherer
Philosophy Department
Bowling Green State University
Bowling Green, OH 43403
(419) 372-7142

BAR CODE APPLICATIONS IN AN ACADEMIC ENVIROMENT

Charles A. Schneebeck
Director of Campus Computing
and
Co-Nghiep Ho

ABSTRACT

Many of the management tasks involved in the operation of college computer laboratories are labor intensive, inefficient and result in low morale on the part of the staff. Bar code technology is used at Fullerton College to manage the collection and reporting of positive attendance in a variety of college laboratories. In addition, the system is used to generate time sheets for hourly employees and as an in-lab software check-out system.

INTRODUCTION

The advantages of bar code technology in the business environment have been clearly demonstrated. Characteristics of speed, accuracy, reliability and the generation of timely data have increased productivity, thereby improving the profit margin. Public educational institutions, in contrast to business, do not measure their efficiency in terms of profit. Efficiency is measured by the quality and quantity of services that can be offered to the community while operating on a fixed budget. Bar code technology has been used to increase efficiency at Fullerton College.

Fullerton College is a community college located in Fullerton, California. The college has an enrollment of 17000 students and offers both academic transfer courses and technical programs. A number of teaching laboratories at Fullerton College are funded on a positive attendance system. The college is required to keep records of the time each student spends in the laboratory. The state then pays the school for the actual time spent.

A PILOT PROJECT

In the past, data for positive attendance were kept using a sign in sheet or a time clock. The time clock worked better because it was harder for the students to "create" times they were in the laboratory. Both systems required a great deal of personnel time to calculate the hours spent in the lab and to prepare the necessary reports. These approaches resulted in a significant number of mistakes in reporting and were generally disliked by the individuals who were responsible for the task. By mid 1985 it became clear that bar code technology might be used to keep positive attendance records more effectively. One of the college microcomputer labs was chosen to run a pilot project. The lab serves approximately one thousand students per semester for a total of about fifteen thousand student hours. In order to implement a system it was necessary to choose the hardware for the job, select the type of bar code to be used and a method for issuing the bar code to the student, and to develop a database program to manage the data and generate reports.

For hardware a PC/XT clone with a 20MB hard disk drive, 640KB of RAM, a wedge bar code reader and a dot matrix printer were chosen. dBase III Plus was used for the database and Code 39 was chosen for the bar code. These materials were used to gather data on positive attendance and generate the necessary reports.

Code 39 high density adhesive labels covered with polyester were purchased from a commercial vendor. A bar code label was attached to a bar code label was attached to a paper "ID" card that was issued to each student at the time the student was entered into the database. This "ID" became necessary for a student to use the lab facilities.

A relational database was developed which consisted of a "student" database with each student's name, middle initial, date of birth, class number and bar code identification number. The necessaty information was keyed into the student database by a lab attendant when a student used the lab for the first time each semester. The screen used to prompt the lab attendant was similar to the following:

WELCOME TO FULLERTON COLLEGE
MICROCOMPUTER LAB

INITIAL ENROLLMENT SCREEN

Please enter information below

Last Name :
First Name :
Middle Initial : **Bar Code ID #**(use wand):
Date Of Birth :
Class Number :

A second database contained the student bar code ID number and was used to record the dates and times that the student used the lab. The date, time in and time out were written to the database from the computer calendar and clock when the student used the wand to read his bar code ID.

REPORTS

A variety of reports was generated from the database. A record of total student hours of lab use was prepared for the state. A few selected teachers were given weekly or accumulative reports on the number of hours the lab had been used by their students. Time cards were printed at the end of each semester to be used as documentation in the event of an audit by the state. Because the reports were prepared electronically, it was no longer necessary for lab personnel to spend many hours per week preparing them manually. The morale of the staff was improved and the hours saved by the system were used to provide tutoring to students who were using the computer resources in the lab.

Collection of data using bar code entry helped produce reports that were virtually error free. Mistakes that did occur were made by the lab attendants when they were keying information into the student database by hand. Having a system that could collect data rapidly and accurately helped to increase the number of student hours to be reimbursed by the state, and resulted in additional income to the college. Security in the lab was somewhat improved. Every student entering the lab was required to "clock in" by reading the bar code from the lab "ID" card. If the bar code that was read was not included in the database, the computer sounded an alarm and the following message appeared on the screen.

WARNING

YOU HAVE NOT BEEN ENTERED IN THE
COMPUTER DATABASE !!
PLEASE ASK A LAB AID TO ENTER YOU NOW

With relatively small amount of programing, it was possible to expand the uses of the system to include a time keeper for hourly employees and a check out/check in system for software, documentation and other materials that are used in the lab.

The time keeper allows the employee to read a bar code menu to activate the program. Then the employee's ID bar code is scanned. The date and time are written to the database. The save sequence is used at the conclusion of each work period. At the end of each month a report is generated that shows the times worked and calculates the pay. (Fig. 1). This approach has all of the advantages of a time clock with the additional advantage of preparing a time sheet for payroll. The need to calculate the time sheets by hand has been eliminated, and a report can be generated for administrative purposes at any time during the pay period.

A check out/check in system for software, documentation and other materials which are borrowed by students for use in the lab has been an extremely effective means to reduce the loss of these resources. A relational database uses the student database and an "items" database. The items database contains all of the items that can be borrowed by students for use in the lab. Each item is identified with its own bar code identification tag. When a student wants to check out an item the lab attendant scans the check out bar code from a bar code menu. The student is then asked to scan his bar code ID.The computer checks to see if the student is in the database. The student may not use the materials until he has been entered into the database. If the check confirms that the student is actively enrolled, the bar code of each item to be checked out is scanned. A list of the items is displayed on the monitor and written to the database. At check-in time, the "check-in" bar code from the bar code menu is scanned. Then the bar code from each item is scanned. When all of the items have been scanned, a "check out complete" barcode from the menu is scanned. The computer performs a check to see if all of the items have been returned. If so the student is clocked out. If all of the items have not been returned, a list of the missing items will be displayed on the monitor.

The bar code checkout system has virtually eliminated the loss of materials that are loaned to students in the lab. Individual parts of a documentation package can have their own bar code identification which increases the likelihood that templates, quick reference guides and other small items will be returned. The system is fast and easy to use. It generates data that are useful for more effective management. For example, it is possible to have reports that show use patterns for lab materials.

By the end of the semester it was clear that the new system had many advantages over the previous methods of record keeping, and that the key to its efficiency was the use of bar code technology. Without bar code technology the system would have been much too slow with an unacceptable error rate.

STUDENT IDENTIFICATION CARDS

A problem with the system was that every student who used the lab had to be issued a new "ID" card each semester. The problem became more apparent as we considered expanding the system to additional labs. A check showed that there were at least seven different "ID" cards being issued independently by different labs, the library, and student government. The library was in the process of preparing for a bar code inventory system and had started the development of a student ID bar code. With the cooperation of the District Information Services, which runs the mainframe computer, and Admissions and Records, it was possible to pull each student's name from the college database, combine it with the student permanent number and print the information with code 39 bar code on a student ID card. The cards were printed in-house using a laser printer. It was necessary to laminate the front of each card to keep the bar code from coming off when in contact with the plastic holders in wallets. Seventeen thousand cards were prepared at a cost of less than three cents per card. With some minor adjustments the cost of each card will be 0.4 of a cent. The new ID card eliminated the necessity to buy bar code or to prepare a new card for each student each semester. They were issued to students in the library and were accepted as identification in most of the labs and the library. However the ID cards would have been much more useful if they had included a picture of the owner.

BAR CODE NETWORKING

The pilot program demonstrated the desirability of using bar code as a management tool in the microcomputer labs. Other departments on campus became aware of the system and wanted to implement the technology in their labs. As a result, bar code equipment has been installed at nine different locations with a total of fourteen input devices. To use the same strategy as the pilot project it would have been necessary to use fourteen complete systems, an unacceptable option. Instead we decided to use a networked bar code data collection system. The bar code data collection system is a networking environment using bar code data only. It will collect data from different locations simultaneously, validate data and transfer data to the host computer (Zenith 248). Starnode Barcode Terminals are used for data input. The system has two functional parts; data collection and processing.

DATA COLLECTION

The system organization is described in Fig. 2. The host computer is operated under an interface program we call Yasin's User Interface (YUI), and the terminals are operated under the Prompt Control Language program (PCL). The PCL program prompts the user for bar code input, checks for appropriate bar code type and length, validates input, displays error notices if any and performs a variety of internal functions such as invoking counters, combining many bar codes into one message, etc. Messages are then sent to the host computer together with the terminal number, terminal location and message length.

The YUI program analyzes the incoming message, validates the message, sends feedback to the terminal, constructs a data base from those messages and saves the raw data to disk. It can also send and receive ASCII text messages to and from terminals, down load PCL programs to terminals and perform other house keeping functions.

The primary task of the YUI program is to interact with the PCL from the terminals to generate a log file. The log file contains the location from which data was entered, the student I.D. number, a prefix code, class number, clock in time and clock out time. When a user scans his I.D. the message is sent to YUI which checks to be certain it is a valid user. If so a confirming message is sent to the PCL and the user data including a user prefix code is written to the log file. If the user is not registered, YUI will send an error message to the PCL and access to the lab is denied.

When an employee clocks in or out, he must scan an employee prefix barcode before scanning his I.D. card. The prefix signals YUI to verify the input is coming from an employee. Then the time in or out is written to the log file with the prefix code for employees. A similar procedure is used for checking out items and returning them to the lab.

DATA PROCESSING

In order to generate reports we use a relational database which consists of two parts. The first is the User Profile which contains the student I.D. number, name, birthday and class number. The second is the User Log File which contains only the student I.D. number, time in, time out and location of input. The User Log File is generated from YUI's raw data log file.

Data for the User Profile is entered on a separate PC and transferred to disk storage on the host computer. When the User Profile is updated, a subset called the RAM Base is generated. The RAM base is resident in a RAM drive and contains only location, user I.D. and class fields. The small size of this subset (32 bytes/user) significantly speeds the process of verifying registered users by YUI.

dBase III is used to generate reports from the relational databases. The time spent in the lab by each student in a class is prepared for instructors. (Fig. 3). Lab use profiles can be generated daily or weekly as needed by management. (Fig. 4). Employee time sheets are generated once a month. (Fig. 1). Check in/check out information can be prepared on demand.

The YUI screen (Fig. 5) allows the operator to have instant information on the number of students using the labs at any moment in time. It also indicates the number that have checked in and have checked out during the day.

By using bar code entry, lab managers can request specific reports to be prepared or send other messages. The most recent students and items to be checked out or in are shown on the screen and information regarding computer use is available for the operator.

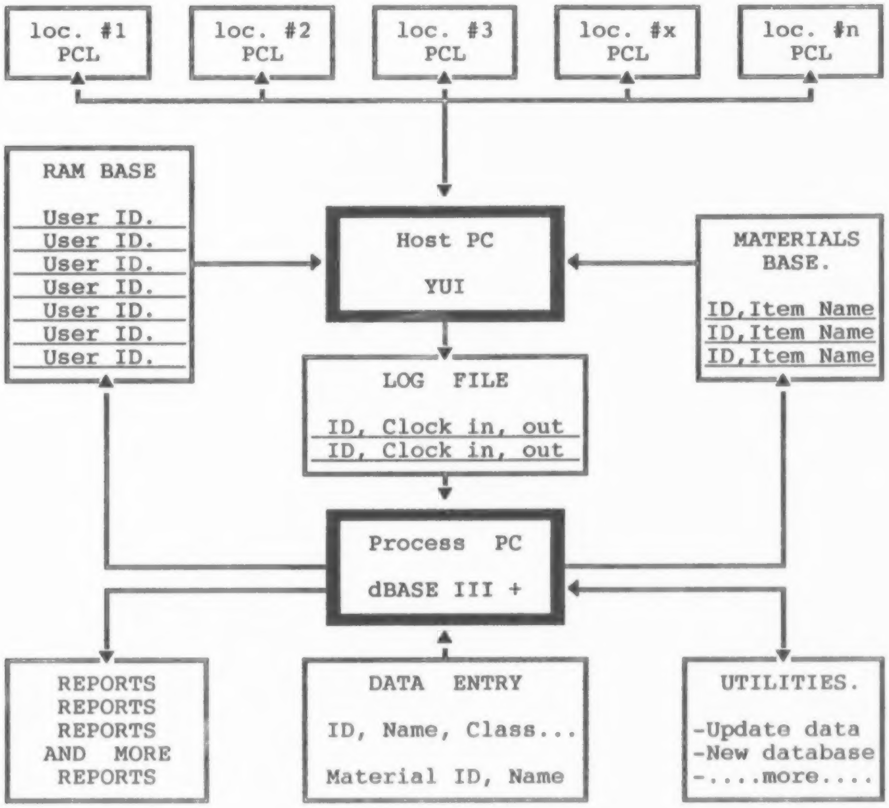
The bar code data collection system makes it possible to give teachers information regarding student use of learning resources. It has been used effectively to reduce the loss of software and documentation that is checked out for student use in the labs. It has reduced the paperwork needed for preparing employee timesheets, and is a valuable tool for lab managers in deciding on hours to keep the lab open and staffing patterns.

In conclusion, bar code technology has been used at Fullerton College as an effective tool for teachers, students and managers.

Charles Schneebeck
Fullerton College
321 East Chapman Ave.
Fullerton, CA 92634
(714) 871-4221

```
name:  Jones, Bill
social security no:  555-88-9878    rate:  $5.50     approved: ____________________
school & dept:  Fullerton College ,  Instruction   budget charge: ____________________
date:
       to: 05/01/87    from: 05/31/87       employee signature: ____________________

     date     hrs     in        out          date     hrs     in        out
   05/04/87 1.00 12:00:03 12:52:57 * 05/13/87 1.00 11:58:14 12:54:33 *
      .       .       .         .  *     .       .       .         .  *
      .       .       .         .  *     .       .       .         .  *
      .       .       .         .  *     .       .       .         .  *
   05/12/87  2.00 09:50:43 11:54:09 * 05/21/87 4.00 08:52:24 12:52:34 *

   total hours: 34.00      rate of pay: $5.55          total due: $200.00

I hereby certify that work and hours specified on this time card were satisfactorily per
formed by the employee signing the card and a total of 34.00 hours is correct.

                                supervisor: ______________________________
```

(Figure 1) Employee Time Sheet


loc. #1 PCL
loc. #2 PCL
loc. #3 PCL
loc. #x PCL
loc. #n PCL
RAM BASE
User ID.
User ID.
User ID.
User ID.
User ID.
User ID.
User ID.
Host PC
YUI
MATERIALS BASE.
ID,Item Name
ID,Item Name
ID,Item Name
LOG FILE
ID, Clock in, out
ID, Clock in, out
Process PC
dBASE III +
REPORTS
REPORTS
REPORTS
AND MORE
REPORTS
DATA ENTRY
ID, Name, Class...
Material ID, Name
UTILITIES.
-Update data
-New database
-....more....

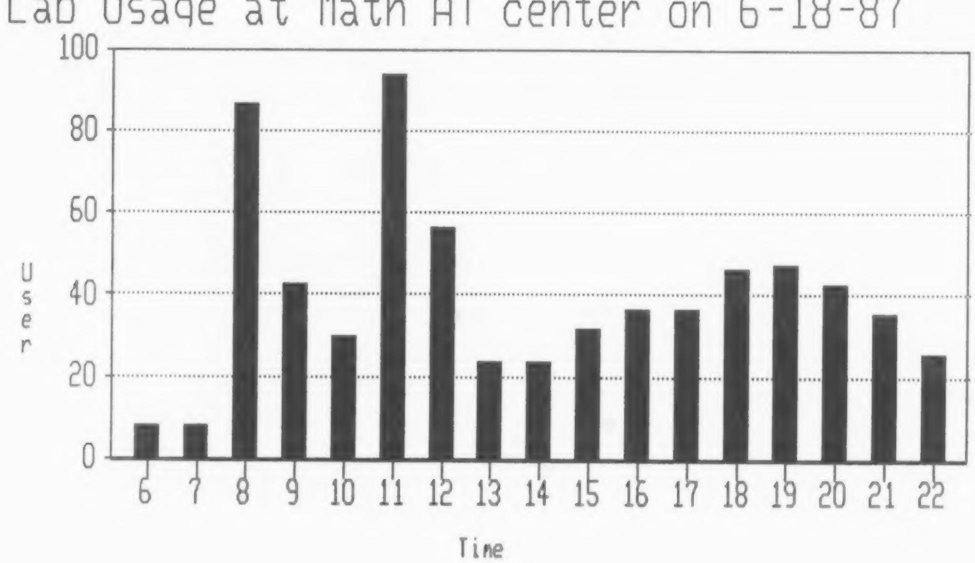
(Figure 2) System Organization

CLASS LAB USAGE REPORT
FOR ROOM 611, MATH AT CENTER

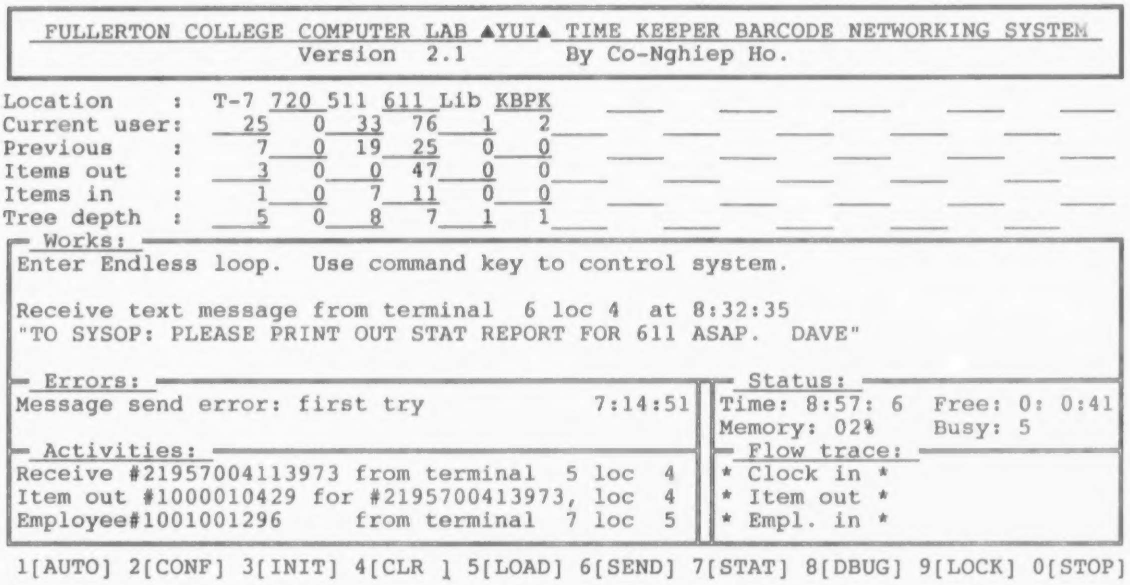
TICKET # COURSE NUMBER AND TITLE INSTRUCTO R

	week1	week2	week3	week4	week5	semester total
BHAIJEEWALA, NWZHAT	1:13	0:00	0:00	0:00	0:00	1:13
BROERSMA, MENNO D	0:50	6:00	0:00	0:00	0:00	7:00
CHANNEL, TODD D	0:00	0:00	0:00	0:00	0:00	0:00
COLLIN, CHRIS L	0:00	0:00	0:00	0:00	0:00	0:00
.	.	.	.	.	.	.
.	.	.	.	.	.	.
SU, WAYNE C	0:00	2:12	0:00	0:00	0:00	2:12

(Figure 3) Student Lab Usage Report



(Figure 4) Lab Use Profile



```
  FULLERTON COLLEGE COMPUTER LAB ▲YUI▲ TIME KEEPER BARCODE NETWORKING SYSTEM
                     Version  2.1        By Co-Nghiep Ho.

Location     :  T-7 720 511 611 Lib KBPK
Current user:    25   0  33  76   1    2
Previous     :    7   0  19  25   0    0
Items out    :    3   0   0  47   0    0
Items in     :    1   0   7  11   0    0
Tree depth   :    5   0   8   7   1    1
 Works:
Enter Endless loop.  Use command key to control system.

Receive text message from terminal  6 loc 4  at 8:32:35
"TO SYSOP: PLEASE PRINT OUT STAT REPORT FOR 611 ASAP.  DAVE"

 Errors:                                     Status:
Message send error: first try        7:14:51 Time: 8:57: 6  Free: 0: 0:41
                                             Memory: 02%    Busy: 5
 Activities:                                 Flow trace:
Receive #21957004113973 from terminal  5 loc  4 * Clock in *
Item out #1000010429 for #2195700413973, loc  4 * Item out *
Employee#1001001296      from terminal  7 loc  5 * Empl. in *
 1[AUTO] 2[CONF] 3[INIT] 4[CLR ] 5[LOAD] 6[SEND] 7[STAT] 8[DBUG] 9[LOCK] 0[STOP]
```

(Figure 5) YUI Screen

The Application of Technology to a Basic Photography Class

Merton E. Thompson, University of Wisconsin-Stevens Point

ABSTRACT

This paper traces a course in basic photography from traditional delivery systems to the implementation of microcomputer authoring system and videodisc delivery. Included are advantages and frustrations at each of the steps along the way. The paper concludes with a brief discussion of future directions and applications.

Like many university classes, the basic photography class that I teach at the University of Wisconsin-Stevens Point began with a traditional approach to the delivery of the information. There was heavy dependence upon textbook readings by the students; classes consisted of lectures supported by visuals; and grading was based upon a combination of exams and students assignments.

This system had a variety of advantages, particularly for the instructor. Class preparation time was kept to a minimum and consisted mainly of checking to determine the topic of the day, pulling the appropriate notes from the file, checking to see that the slides and transparencies were ready, and walking into class and talking about the topic for 50 minutes. Other preparation consisted of staying abreast of current developments in the field and updating topics as necessary.

One of the first indications that there were flaws in the system was in the quality of the student assignments, particularly the early assignments in the class. While some students passed in assignments of excellent quality, others turned in assignments that did not meet even the most basic criteria. Observations of students working on the projects, and discussions with them demonstrated that this range of quality was not attributable entirely to student motivation and interest in the class. It was often the students turning in the poorest assignments who were putting forth the most effort and asking the most questions during and after class.

A survey of the students enrolled in the class indicated that there was a wide range of experiences and knowledge of photography. Some students had no previous experience in photography in general while others had considerable experience in 1 or more specific topics. In most cases the students with experience lacked information in specific topics covered by the course or had gaps in their understanding of the subject and its applications. These findings were reinforced by observing student behavior during lectures.

Determining how to deal with this range of student knowledge became the focus of my attention. It soon became evident that a change in the pacing of the information presented in class would not sufficiently solve the dilemma. Slowing the pace to ensure that all understood each concept before moving on to the next tended to produce comments of "boring" or "too much time spent on small details." Speeding up the delivery of information left more of the students confused and frustrated.

The logical step seemed to be to provide supplemental materials that students could use as appropriate to help themselves understand and apply a particular concept or topic. The decision was made to produce a series of self-instructional booklets covering the procedures and concepts which were taught during the first part of the class. A series of booklets covering such topics as loading film into developing tanks, developing film, and operating an enlarger were designed. These booklets were then produced using photographs of the facility in combination with printed information.

Student reaction was immediate and positive. The booklets were of great assistance to inexperienced students because they now had immediate access to the specific information needed. More experienced students quickly perused the information to ensure that they were familiar with all the steps or concepts involved. As a result more class time was devoted to an in depth study of photography and to expanding the experience and knowledge base of all the students.

However, limitations of the booklets soon became evident. First, the booklets were very labor intensive to produce. Based upon class size it was determined that 4 or 5 copies of each booklet would be necessary. This was too small a number for any type of printing service, so each booklet was put together individually, printing the photographs and dry mounting them next to the text, then laminating the pages and assembling the booklet. Nearly as soon as the booklets were assembled, the need for revisions became necessary because of changes in procedures or equipment in the lab. Revising the booklets became an annual event and often amounted to virtually starting over.

A second limitation was the lack of feedback provided to the students as they worked their way through the booklets. It was relatively easy for less motivated students and those working against the pressure of the clock to assume that they understood and could successfully use the concepts and procedures in the booklets. It was often after an assignment was completed and evaluated that it became obvious to the student that he/she did not have a full grasp of the information. Without an immediate check on comprehension, it was easy for students to be misled into believing that they understood the material and could apply it appropriately.

While the booklets were a step in the right direction, it was clear that they did not satisfactorily solve the problems. At about this time, I was developing a knowledge of the applications of the microcomputer to instruction. It seemed that the microcomputer might be of assistance in solving some of the problems associated with the class. Therefore, the use of a simple authoring system in the class was begun. The authoring system chosen Study Guide allowed the writing of multiple choice, matching, and fill-in-the-blank questions covering the content of the course. The students, in turn, worked their way through the questions as their schedules permitted and as topics were presented in class. Each question was categorized into a topic area of the course. Each topic introduced in the class had an accompanying series of questions that covered the major concepts of the topic.

Students were encouraged to work through the questions after each topic was introduced and discussed in class. The program provided immediate feedback as the students answered the questions. Questions answered incorrectly were repeated by the program. The program also kept score for the students and those who completed less than 80% were prompted to repeat the questions at another time. Records were also maintained for the instructor on each of the students and their scores.

This addition was also well received by the students. Many students enrolled the class had had no previous experience with a microcomputer. But most found this introduction a pleasant one and they quickly overcame any initial fears of the computer. Experienced microcomputer users also found the application useful and helpful assistance in the class.

Students particularly enjoyed the ready availability of the information and the ease with which they could work their way through the program. They also mentioned the fact that they knew immediately when a question was answered incorrectly and could go back through the questions as often as desired. Additionally, students mentioned the learning environment of the computer, which they found to be a non-threatening way to check on their understanding of the photographic information. College students, as students at other levels, are often reluctant to offer an answer to a question publicly if they believe they might be wrong. The computer allowed them to offer answers without the threat of public humiliation.

As the instructor, I found the record keeping function particularly helpful in following up on students. When I noticed a student's name not listed in the records of the computer, I made a point of discussing with the student any problems that might be present. I found there was a high degree of correlation between the amount of time spent answering the questions on the computer and test scores, as well as with the quality of assignments.

Another feature that was especially helpful was the ability of the computer to randomly distribute possible answers to the multiple choice questions. This helped to eliminate the possibility of students memorizing the letter of the correct answer. The questions were also randomly presented to the students within a given topic area. This helped keep the boredom factor to a minimum when a student needed to go through a topic area several times.

However, after a few weeks of using the authoring program, a number of limitations, with this system as well began to become obvious. Although the program did allow the writing of a variety of types of questions, several of the possibilities had severe shortcomings. For example, fill-in-the-blank questions were limited to 5 possible correct answers and the student had to enter one of the 5 possibilities exactly, including the use of upper and lower case letters. In many cases when this type of question was used, students who understood the concept received a response of incorrect from the program because of slightly different phrasing of their answers or a variation in the use of capital letters. The students found this to be very frustrating. For this reason, fill-in-the-blank questions were abandoned even though in many cases I would have preferred it from a teaching point of view.

The use of multiple choice questions also faced limitations. In this case, the possible answers were limited to 30 characters. This led to the use of phrases and abbreviations which sometimes caused confusion for the students. Once again the result was sometimes the testing of semantics as opposed to concepts of photography.

The overriding limitations, however, were the inability of the program to display any type of graphics and the lack of any branching based upon student responses within the program. Photography is a visual subject and the concepts need to be seen as well as written about. The inability to use any type of graphics greatly limited the types of information that students could test themselves on. The lack of branching within the program meant that it was severely limited in its ability to actually present information to the students. This program is designed to simply display a series of questions to the students and provide a brief statement of feedback on the correctness of the student response.

For these reasons, a search for more powerful software was undertaken to expand the use of the computer in delivering information to the class. The advantages mentioned above to the students were too valuable to discard due to the simplicity of a particular program. The problem became one of finding software that overcame most or all of the shortcomings while maintaining the strengths. The software that most fully met this criteria is a complete authoring system called Quest. This system is designed to permit the easy production of computer assisted instruction. It allows branching from one frame to another. Students correctly answering a series of questions concerning one concept are branched to a second concept, while students responding incorrectly are branched to a more detailed explanation of the information complete with visual information. This allows me, through the use of the computer, to begin to fully deal with students as individuals.

The use of visual information mentioned earlier is accomplished in three ways. First, the program contains a shape editor so that computer graphics images can be easily created and entered into a sequence. Second, the program will also import graphics images created by other microcomputer software. In addition, the program will control a video disk player and therefore provide access to high quality images similar to the quality of the photographic images the students produce in the class. In addition, graphics and text can be superimposed on top of a videodisc image to point out specific characteristics of the image. Working with this feature allows the use of almost any videodisc image to teach concepts in photography. Once an example of a particular concept is located it is fairly easily incorporated into the instruction contained on the computer.

The major limitation to the use of the program has been the complexity of creating the instruction. Because of the large number of variables built into the program, it is time consuming to develop an instructional unit. For this reason, I began by using the program only in a limited fashion. The first step was to translate the questions from the previous software to this more complete authoring system. This provided the opportunity for me to develop my experience with the software and become more knowledgeable about it. In this process, I was able to eliminate most of the shortcomings mentioned earlier concerning the simpler authoring program.

Currently I am expanding the use of the branching by beginning to write instructional sequences. I have begun with remedial sections for students unfamiliar with a particular concept such as film developing and depth of field. In these applications, I find the use of videodisc images particularly useful.

Within the next year, I expect to have a major portion of the beginning concepts of the photography course taught by the computer. This will allow me to spend more of the class time working directly with the students and their specific needs instead of lecturing to all as if they had exactly the same amount of background and experience with the topic.

This photography course has undergone a large number of changes over the last several years. The application of technology, whether in the form of instructional slides or microcomputer controlled videodisc has the potential to greatly improve the science of teaching. However, care must be taken to ensure that the technologies chosen are appropriate for the students and the content of a particular course. Above all else, the recent developments in technology demonstrate that there are a wide variety of successful teaching new concepts.

References

Quest. Salt Lake City, Utah: Allen Communication, 1986.
Study Guide. St. Paul, Minnesota: Minnesota Education Computing Corporation, 1984.

Merton E. Thompson
438 COPS Building
University of Wisconsin-Stevens Point
Stevens Point, WI 54481
(715) 346-4426

IS THERE AN EXPERT SYSTEM
IN YOUR IV/A FUTURE?

Dr. Greg Baur

Western Illinois University

Abstract

There is much interest and discussion today about artificial intelligence and, in particular, expert systems. What are expert systems? How can they be combined with IV/A systems to produce an enhanced instructional delivery system? The paper will examine answers to these questions and give an example of such a system.

Introduction

In recent years, research in artificial intelligence has had many important successes. One of the most significant of these successes has been the development of very powerful computer systems which have come to be known as "expert systems" or "knowledge-based systems". These programs are designed to solve problems by using factual knowledge of specific areas of expertise. For example, systems have been developed to diagnose diseases, configure computer systems and prospect for the earth's minerals at performance levels equal to or surpassing those of human experts. The potential of these systems to capture rare or expensive human knowledge has led developers throughout the world to look for new ways to apply this technology.

What Are Expert Systems?

In this section, we will discuss the nature and structure of expert systems. It will also discuss how expert systems are organized, how they differ from conventional programs and the kinds of tools used to build expert systems.

The Nature of Expert Systems

Basically, an expert system is designed to simulate the behavior of a human expert as he or she solves a given problem. To perform this simulation, the factual knowledge and reasoning techniques used by the human must be captured and converted into a representation that is understood by the computer.

When expert systems were first being developed, the process of extracting needed information from the human expert was considered by most people to be an artistic rather than a scientific one. Today, the process is much better understood and, with the development of powerful system building tools, the result has been an increasing number of problems that are being solved by expert systems.

The process of building an expert system is commonly called knowledge engineering. This process involves interaction between one or more human experts and a knowledge engineer. The task of the knowledge engineer is to extract facts, rules and strategies from the human expert(s) and to put this information into a form understood by the computer system. The process is often a long and tedious one as you can probably imagine since the knowledge engineer must strive for completeness and consistency of information.

The result of the knowledge engineering process is a computer program which behaves in an intelligent and skilled fashion. The program is able to solve a given program in the same manner as a human expert.

To better understand the nature of expert system technology, let's compare human expertise and artificial expertise. There are good and bad features of both expertises.

Artificial expertise is permanent and easy to transfer. Human expertise is perishable (death, retirement or resignation) and is difficult to transfer as we pointed out in our description of knowledge engineering. Artificial expertise is easy to document and produces consistent results while such is not the case with human expertise. Finally, while the cost of building an expert system is high, artificial expertise is still much more affordable than human expertise.

But, artificial expertise has several significant limitations. Human expertise is creative and adaptive in its problem solving approach which is not the case with artificial expertise. Artificial expertise is neither creative nor adaptive and cannot benefit from sensory experiences. Finally, the knowledge used by artifical expertise is technical and limited to what it has been told whereas human expertise has the power of common sense knowledge.

Even with these limitations, artificial expertise has demonstrated, through expert systems, that it is a viable alternative to human expertise. The performance of artificial expertise will continue to be enhanced as the technology grows.

Expert System Organization

As we examine the structure of an expert system, we find that it has two basic parts, the knowledge base and the inference engine. The inference engine acts upon the knowledge base to deal with the problem to be solved.

The knowledge base is made up of the facts (data) and rules (or other representations) that describe the knowledge of the domain (problem) area. An example of a knowledge base is shown here.

FACTS:	Johnny has test average of 55 and has missed over 50% of the classes this term.
RULE:	If a student has a test average less than 70 and has a poor attendance record, then the student has a problem that I need to deal with.

The information in the knowledge base is used for making decisions relative to the solution of the problem to be solved.

The inference engine is the other main component of the expert system. It contains an interpreter which decides how to apply the rules to infer new knowledge and a scheduler that decides the order in which the rules are to be applied.

Some problems arise with the implementation of this structure when building an expert system. It is a fairly straightforward task to take the known facts and rules and put them into the desired representation. It is much more difficult to build the inference engine because there is no simple, general way to characterize it. Its structure depends on the problem to be solved and on the way that the knowledge is represented in the system. Some expert system building tools provide a shell which contains the inference engine and the system builder needs only to instill the knowledge base. One difficulty with this technique is that the shell lacks flexibility so that only a very narrow range of problems can be solved. The other alternative is to use a high-level language such as LISP, PROLOG or Pascal; more will be said on this topic in a later section.

We have mentioned in this section that knowledge may be represented in several ways to the computer. The most commonly used method of representating knowledge is the rule. It is popular because rules are easy to write and interpret. The form is the familiar IF-THEN statement with multiple IF antecedants and one or more THEN consequents. An example was given earlier in this section. Semantic nets and frames are two other techniques for representing knowledge.

Expert Systems vs. Conventional Programs

Another way to help understand the nature of expert systems is to compare them with conventional computer programs. The basic difference is that expert systems manipulate knowledge while conventional programs manipulate data.

It is common to have the terms knowledge and data used interchangably. However, the terms are not synonomous as they apply to expert systems. A conventional program uses data (facts) and performs a series of predetermined processes (rules) that are the same for all sets of data. On the other hand, expert systems use known facts (data) and inferred facts (from rules) to solve the problem. The use of the rules will vary according to the data.

Another basic distinction between expert systems and conventional programs is that conventional programs are algorithmic and expert systems are hueristic. An algorithmic program uses a formal procedure that is guaranteed to produce a correct or optimal solution. Such is not the case with a hueristic (rule of thumb) program. Hueristics are ways of limiting the search for solutions in domains that are difficult and poorly understood. Because expert systems use hueristics, they can make mistakes and thus cannot guarantee a correct or optimal solution.

It is not a question of whether you should use an expert system or a conventional program. As you can see from the previous discussion, each is designed for a different purpose. It is often feasible and desirable for both to be used in the same application with each doing specific tasks. In these cases, it will likely be the expert system that controls the use of the conventional program.

Types of Expert System Building Tools

We mentioned in an earlier section that a builder of an expert system has two types of tools potentially available. These tools include standard high-level programming languages and knowledge engineering languages or systems.

Programming languages that are used for expert system applications include symbolic-manipulation languages such as LISP or PROLOG and procedure-oriented languages such as Pascal or FORTRAN. There are advantages and disadvantages to the use of these programming languages.

The main advantage is that they allow a great deal of flexibility for the expert system builder. Because of this flexibility, an expert system built with these tools can solve a wider variety of problems within the same domain. The disadvantage is that these languages are more difficult to apply because they give little or no guidance as to how knowledge should be represented or mechanisms for accessing the knowledge base.

Knowledge engineering languages are sophisticated expert system building tools. They can be classified as skeletal systems or general-purpose systems and are comprised of an expert system building language and an extensive support environment.

The advantage of the knowledge engineering language is that it is easier for the user to apply. The language essentially dictates how knowledge is represented and contains mechanisms for accessing the knowledge base. Its disadvantage is that, because it is structured, the range of problems that can be solved in a given domain is limited to the constraints enforced by the language. While the cost of good knowledge engineering languages can be quite high, they probably do not take as much personnel development time which is also a high-cost commodity. While the names of knowledge engineering languages are not as well known as those of standard high-level programming languages, there are a number of good ones available and more are continually being developed.

Combining Expert Systems with IV/A Systems

Now that you have a basic understanding of expert systems, let us look at how expert systems can be combined with IV/A systems to produce an enhanced instructional delivery system. The discussion will center around the creation of systems that can be used for education and/or training applications.

For our discussion, it is appropriate to consider a teacher or a trainer as a human expert. These people have the skill to diagnose the needs of a learner or a trainee as he or she attempts to gain some desired knowledge or skill. Once these needs are assessed, the teacher or trainer designs a sequence of activities (instruction) to meet these needs. The result of the instruction is the acquisition of the knowledge or skill in an effective and efficient manner.

Most IV/A systems contain a large database that is manipulated with a microcomputer. The user interacts with a microcomputer program which accesses the database. The user is able to control how and what information is to be accessed. The result may be an ineffective and inefficient use of the system by the user.

Most computer programs that drive IV/A systems designed for education and training applications do not have much flexibility for the user. The user must move through some predetermined path which may or may not produce the desired end results.

While some current computer programs are quite sophisticated, they fall considerably short of a substitution for the live teacher or trainer. They do not have the capability of making judgements and changes in strategy at the same level of expertise as a human being who is a trained professional educator or trainer.

How can the expert system help solve the dilema? An expert system can be used to drive the IV/A system database by providing the interface between the user and the database. Since the interface is an expert system, it can perform intelligently and provide a learning situation that more closely resembles that of human interaction between teacher/trainer and student/trainee.

For example, a system such as the one described above might look like this. Suppose that we are interested in helping a student learn about basic parts of a computer. We have one or more videodiscs which contain information and illustrations about the basic computer parts. Students come to the learning situation at different levels of knowledge and understanding about computers. We could build an expert system to assess student knowledge level and plan a strategy (or strategies) for the student. The expert system would emulate a teacher who is an expert in computer science. It would have at its disposal the database of the videodisc(s) and its own knowledge base from interaction with the student.

There are a number of people who are developing intelligent tutoring systems which perform in the manner described above. An area of active research in artificial intelligence applications is intelligent CAI.

What makes IV/A databases so attractive for expert system development is their size and multimedia capabilities. The videodisc has already proven its viability, and the compact disk in the form of CD-ROM and CD-I is receiving a great deal of attention by educators and trainers.

Conclusion

In this paper, we have discussed the nature of expert systems and what they are and are not. We also examined how expert systems can be combined with IV/A systems by using the expert system to control the IV/A database.

Is there an expert system in your IV/A future? It's worth an exploration!

References

Harmon, P. and King, D., Expert Systems, John Wiley, 1985.

Waterman, D., A Guide to Expert Systems, Addision-Wesley, 1986.

Greg Baur
Dept. of Computer Science
Western Illinois University
Macomb, IL 61455
(309) 298-1452

USING PATIENT SIMULATIONS TO TEACH PROBLEM-SOLVING

LYNN JOHNSON - UNIVERSITY OF IOWA

DR. MICHAEL W. FINKELSTEIN - UNIVERSITY OF IOWA

ABSTRACT: *Oral Disease Simulations for Diagnosis and Management* (ODSDM) was developed to train dental students in the problem solving skills necessary to diagnose and treat non-tooth related diseases. ODSDM consists of a laser reflective videodisc combined with a computer program. The computer program delivers multiple case studies, while the videodisc presents high-quality visuals. The College of Dentistry has the potential to create 9,000 patient simulations and the program is easily adapted by other health care disciplines.

The ODSDM program consists of five major sections. The first section, the Case Summary, presents the patient. In the second section, Investigation, the student takes a patient history and obtains clinical information. In the third and fourth sections, Clinical Impression and Management, clinical diagnoses are formed and treatment suggested. Both sections require the student to make a series of decisions versus one final judgement. The Critique contains the student's scores and the correct diagnosis and management. An expert describes the information vital to decision making and provides a rationale for each clinical impression and management decision. A printout repeats the Critique and the student's clinical impression and management. This enables students to immediately compare their decision making with an expert's.

THE PROBLEM:

Dental schools adequately present a basic knowledge of soft tissue and bone diseases. Clinical courses in diagnosis and medicine expose students to the principles of diagnosis and management. However, time and access to patients limit students' clinical experiences in applying their knowledge to problems of diagnosis and management. As a result of these limitations students complete their formal training without the problem solving skills necessary to manage non-tooth related diseases.

As information grows and technology develops, dentists need problem solving skills to keep up with these changes and to apply them in diagnosis and patient management (Barnett, 1984).

Patient simulations have been used successfully to teach problem solving in medicine. In the absence of a sufficient quantity and variety of patients, these simulations have enabled medical students to practice diagnostic and management techniques. This was done with no harm to the patient and at a reasonable cost (McGuire, 1976) *Oral Disease Simulation for Diagnosis and Management* (ODSDM) is a series of computer videodisc patient simulations that make a wide variety of dental patients available to students.

INSTRUCTIONAL CRITERIA:

ODSDM has been designed to meet the following instructional criteria. First, students must be given the opportunity to take an active role in gathering information, diagnosing, and managing patient diseases. Students must have the opportunity to practice obtaining a complete history focusing on the patient's chief complaint, medical and dental history, and clinical findings. The students must also practice applying this information to form a clinical diagnosis and recommend treatment.

Second, students must be given the opportunity to develop problem solving skills and apply them to a broad range of clinical problems. This should include practice with both common and unusual diseases and patient histories that students rarely encounter in training, but dentists encounter in practice.

Third, students must be given clinical problems that challenge their developing problem solving skills.

Fourth, students must be presented with actual clinical problems. The absence of a sufficient quantity of appropriate patients requires that information be presented using high quality visuals.

Fifth, the first four criteria should be met at a reasonable cost. To diminish cost, faculty involvement should be limited after implementation.

PROJECT DESCRIPTION:

The Hardware:

ODSDM uses a computer controlled laser reflective videodisc. The student views two television screens and communicates via a microcomputer. One screen displays videodisc images; the other screen displays computer text. The microcomputer allows the student to gather visual and textual information on the patient's disease, to give a differential diagnosis, and to prescribe treatment. The microcomputer also gives feedback about the problem solving process.

ODSDM uses an IBM compatible microcomputer (MS-DOS; 640K), a Pioneer LD-V6000 videodisc player, two Sony PVM-1271Q color monitors, and a printer.

The Videodisc:

The videodisc presents high resolution visuals of the patient's disease. It also offers x-rays and histology images. The videodisc images can be randomly accessed almost instantaneously, and they are durable and easily maintained. In 1985, the University of Iowa produced a prototype videodisc, Visual Databases, which contains about 43,000 images from nine university departments. The Department of Oral Pathology and Diagnosis contributed 600 images of sixty patients. Using these images, the Department of Oral Pathology and Diagnosis, Weeg Computer Center and the University Video Center developed the present prototype program, Oral Disease Simulation for Diagnosis and Management.

The Software:

ODSDM has been designed to allow students to develop and apply their clinical problem solving skills to a wide variety of patients. To date, thirty patient simulations have been created.

Three software components combine to form one patient simulation: the simulation driver program, three baseline files, and individual patient files. The driver program presents information from the baseline and patient files, permits movement within the program, scores the student's work, and controls the sequence of computer and videodisc material. The baseline files contain the information that is common to all patients, while each patient file contains the information that is specific to the individual patient.

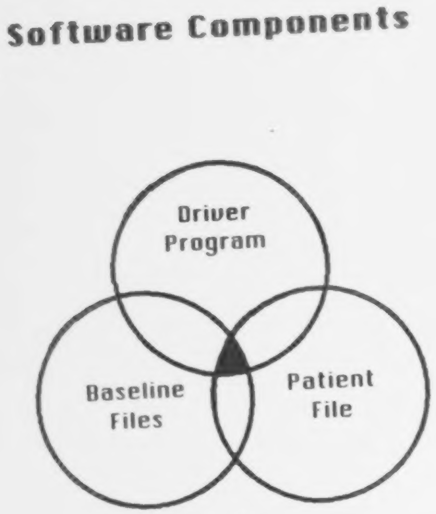


Figure 1.
The three software components of ODSDM.

Each patient simulation contains five major sections: Case Summary, Investigation, Clinical Impression, Management and Critique.

Simulation Map

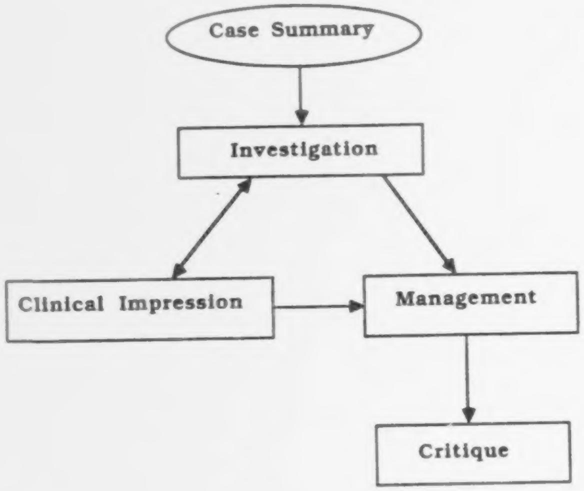


Figure 2.
The five major sections of each patient simulation.

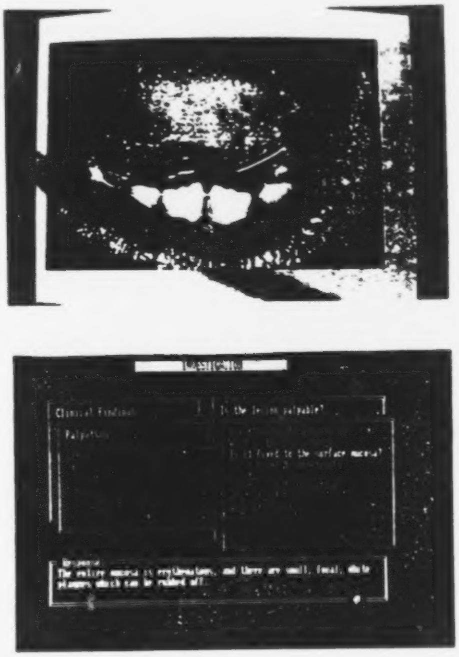
1. Case Summary

The Case Summary presents the patient's chief complaint and the clinical visuals which are accompanied by written descriptions.

2. Investigation:

The Investigation enables the student to gather information about the patient's chief complaint, medical and dental history, and clinical findings. The student selects a question and receives the patient's response. The student may ask all the questions or choose questions that appear appropriate. If the response to the selected question supplies vital diagnostic information, the response comes from the patient file. If the question is not vital to diagnosis a response is randomly chosen from the baseline file based on the patient's age and sex.

Videodisc


INVESTIGATION
Clinical Findings
Palpation

Is the lesion palpable?

Is it fixed to the surface mucosa?

Response
The entire mucosa is erythematous, and there are small, focal, white plaques which can be rubbed off.

Computer

Figure 3.
An Investigation screen demonstrating
a question with its patient response.

Students do not conduct live patient interviews. Instead they select questions from a prepared list. There are three reasons for this: first, typing skills do not create an advantage. Second, research about natural language "interviews" does not prove it superior to other methods (Small, 1980). Third, menus permit questions to be grouped into meaningful categories. Instructional research demonstrates that organization encourages students to learn a systematic approach to information gathering (Ausubel, 1968).

3. Clinical Impression:

Clinical Impression makes use of a decision tree to encourage students to develop an alternative to a "shot-gun" approach to diagnosis. The decision tree encourages the student to make a series of deductive decisions rather than one diagnostic judgement (Vassition, 1984).

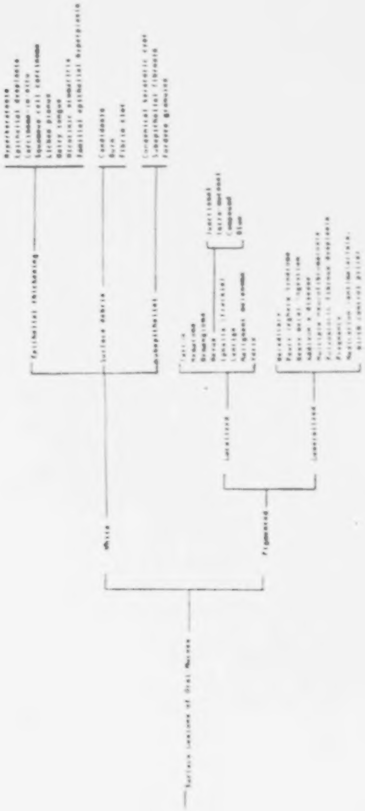

Surface Lesions of Oral Mucosa
White
Epithelial thickening
Hyperkeratosis
Epithelial dysplasia
Carcinoma-in-situ
Squamous cell carcinoma
Lichen planus
Hairy tongue
Nicotinic stomatitis
Familial epithelial hyperplasia
Surface debris
Candidosis
Burn
Fibrin clot
Subepithelial
Congenital keratotic cyst
Subepithelial fibrosis
Fordyce granules
Pigmented
Localized
Tattoo
Hematoma
Hemangioma
Nevus
Junctional
Intra-mucosal
Compound
Blue
Ephelis (freckle)
Lentigo
Malignant melanoma
Varix
Generalized
Hereditary
Peutz-Jeghers syndrome
Heavy metal ingestion
Addison's disease
Multiple neurofibromatosis
Polyostotic fibrous dysplasia
Pregnancy
Medication (antimalarials,
birth control pills)

Figure 4.
A branch from the Clinical Impression decision tree.

Since it is often difficult to make a definitive diagnosis based upon history and clinical findings, the student is allowed to choose up to twelve possible diseases. The most ideal clinical impression is one that contains all the lesions compatible with the information available in the Investigation section.

4. Management:

The student may advance to Management after entering one clinical impression. This emphasizes that a clinical impression precedes patient management.

Management also uses a decision tree to

encourage systematic decision making. The patient's management consists of a series of treatment decisions. These treatment decisions may include a diagnostic test or a consultation with a specialist. Using the test or consultation results, the student must once again consider all treatment options available in the decision tree.

In their management, the student is limited to one definitive treatment. This encourages the student to focus on the most essential treatment.

5. Critique:

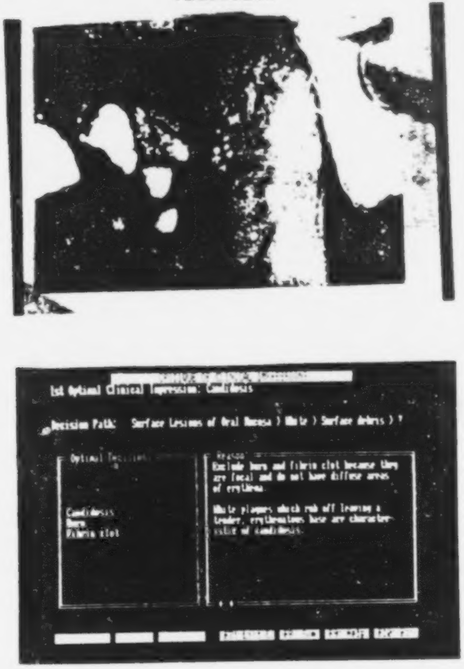
The Critique contains the student's scores, the correct diagnosis and management, follow-up information and visuals, and detailed feedback for each simulation section. In the Critique of Investigation, an expert describes the information that is vital to correct diagnosis and management of the disease. The Critique of Clinical Impression and Management reports the optimal decision path accompanied by an expert's rationale for each decision. The Follow-up describes the treatment outcomes and presents clinical and microscopic visuals of the disease. A printout reinforces the Critique and repeats the student's clinical impression and management decisions. This enables the student to immediately compare their decision making with that of an expert.

Reevaluate Function:

ODSDM is designed to create patient simulations that challenge the problem solving skills of students at various levels of expertise. The novice dental student is not expected to learn correct diagnosis and management skills without guidance. Lacking a broad knowledge base, a young student may incorrectly diagnose and manage a patient. Research suggests that novices need immediate feedback in response to their clinical decisions, while more experienced students learn best with deferred feedback (Gage, 1984). Based on the student's expertise, we turn immediate feedback on and off with a "reevaluate" function.

The "reevaluate" function individualizes any patient simulation for the student. When a novice student selects an inappropriate option, a window with the message "Reevaluate your decision" appears. The inappropriate option and its corresponding tree branch become unavailable. The "reevaluate" function is designed to guide a novice dental student to the correct decision by eliminating incorrect options. A reduced score is their only penalty. The "reevaluate" function is turned off for advanced students. Advanced students diagnose and manage patient's diseases as in real life. Feedback is provided only after completing the patient's treatment.

Videodisc



1st Optimal Clinical Impression: Candidiasis

Decision Path: Surface Lesions of Oral Mucosa) White) Surface debris) ?

Optimal Decision

Candidiasis
Burn
Fibrin clot

Reason

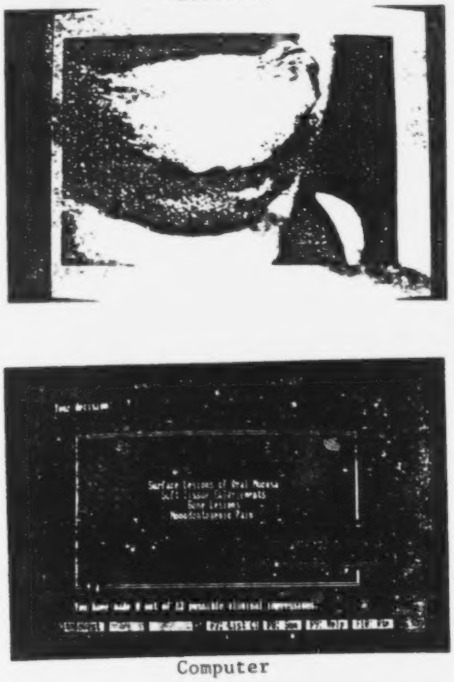
Exclude burn and fibrin clot because they are focal and do not have diffuse areas of erythema.

White plaques which rub off leaving a tender, erythematous base are characteristic of candidiasis.

Computer

Figure 5.
A Critique of Clinical Impression screen presenting the correct decision and its rationale.

Videodisc



Your decision:

Surface Lesions of Oral Mucosa
Soft Tissue Enlargements
Bone Lesions
Nonodontogenic Pain

You have made 0 out of 12 possible clinical impressions.

Computer

Figure 6.
A Clinical Impression screen demonstrating the Reevaluate message.

Creating Patient Simulations:

An author creates a patient simulation by editing a patient simulation template. In the Case Summary the author specifies the patient's chief complaint, age and sex, and describes the clinical visuals. For Investigation, the author writes a response to all patient specific questions. In Clinical Impression and Management the author specifies the correct path through the decision tree and the score for each correct decision. In Management, the author also describes the results for each relevant diagnostic procedure. The Critique requires the author to summarize the vital information for Investigation and to write a rationale for each decision in Clinical Impression and Management.

Defining the Content:

ODSDM may be modified to fit the needs of other health care disciplines. Using a word processor, a faculty member can write questions, responses, and decision trees that fit their specific discipline or philosophy. For example, the classification of oral lesions in Clinical Impression is a clinical scheme unique to the University of Iowa. Another dental school may organize lesions by cause or by basic pathologic process. Modification requires three steps. First, oral lesions must be organized into a decision tree; second, the decision tree must be entered into the appropriate baseline file; and third, new patient simulations must be written or existing simulations modified. A new series of patient simulations may be developed without the involvement of a computer programmer.

FORMATIVE EVALUATION:

During the Fall of 1986, a formative evaluation of ODSDM was conducted to identify areas needing improvement and to measure student attitudes toward the computer videodisc patient simulations. The pilot test examined the design of ODSDM and 20 of the patient simulations.

Thirty junior dental students each completed six to eight patient simulations. During each session with the ODSDM an observer recorded student's questions, problems, and comments. Each student completed a questionnaire and was interviewed.

Student reactions were positive. They found ODSDM easy to use and expressed satisfaction with the quality of the visuals and with the organization of the material. Students consistently affirmed that the Critique's step-by-step rationales for all diagnostic and management decisions was useful. Students also requested increased access to the patient simulations and the representation of more complex patient problems. Students perceive that the simulations will help them develop systematic problem solving skills appropriate to dentistry.

Based on the results of the formative evaluation improvements have been made in the interaction between the student and the computer and in the scoring system. Students identified Management as needing revision. Presently, it does not permit multiple management decisions, nor does it allow a student to prioritize a sequence of decisions. Modifications are being made to correct these deficits.

THE FUTURE

The College of Dentistry and the University Video Center are producing a laser reflective videodisc dedicated entirely to dental images. It will contain approximately 80,000 still images, representing over 9,000 patients from eleven departments. Histology and radiograph databases will also be included. A computer index of each image is being designed for the videodisc.

The Department of Oral Pathology and Diagnosis, the College of Dentistry and Weeg Computing Center are also developing a prototype computer network system to assign patient simulations to students and to provide grading reports to faculty. Other departments within the College of Dentistry, the Department of Sports Medicine, and the Colleges of Pharmacy and Medicine are modifying and expanding the ODSDM to meet their own instructional needs.

The overall purpose of this project is to develop a series of patient simulations that will allow dental students to practice applying the problem solving skills necessary to manage patients and to adapt their skills to meet the needs of their future patients. To determine if ODSDM is meeting this goal a summative evaluation is planned to measure the expected increase in the information gathering, diagnostic, and management skills of present dental students. A longitudinal study to determine the ability of practicing dentists to adapt to changes in the profession is under consideration.

ACKNOWLEDGEMENTS

The authors wish to express their appreciation to Les Finken, Weeg Computing Center, the University of Iowa, for his valuable contributions to the design and programming of *Oral Disease Simulations for Diagnosis and Management*. We also wish to thank Jeff Westfall for his help in preparing this manuscript.

REFERENCES

1. Barnett, O. Computer-based simulations and clinical problem solving. Med Inform 1984; 9(3/4): 277-279.

2. McGuire, C. Simulation technique in the teaching and testing of problem solving skills. J Res Sci Tchng 1976; 13(2): 89-100.

3. Small, DS, Weldon, LJ. An experimental comparison of natural and structured query languages. Human Factors 1983; 25(3): 253-263.

4. Ausubel, DP. Educational Psychology: A Cognitive View. New York: Holt, Rinehart, and Winston, 1968.

5. Vassitiou, Y. (ed.) Human Factors and Interactive Computer Systems. New Jersey: Ablex Publishing Corporation, 1984.

6. Gage, NL, Berliner, DC. Educational Psychology. Boston: Houghton Mifflin Co., 1984.

Lynn Johnson
Department of Oral Pathology
and Diagnosis
University of Iowa
College of Dentistry
Iowa City, Iowa 52242

(319) 335-9656

DESIGN PARAMETERS FOR VIDEO/GRAPHICS OVERLAY APPLICATIONS

D. V. Pigford

Western Illinois University

Abstract

The purpose of this paper is to present design criteria for maximizing the effectiveness of graphics images in a video overlay application. The delivery system of the application is assumed to consist of a level three videodisc enhanced with the ability to combine microcomputer generated images with frames from the video source. The combining of these two graphical images is on one display screen.

Since the production costs of a videodisc are high, the designer's decision to embed a predetermined image on the videodisc or to use computer generated graphics alone, or a combination of the two image sources, is a critical decision, both for cost and effectiveness of the application. Moreover, since the videodisc frames are unchangeable, except for the pseudo updating with an overlay graphics system, this design decision on image placement is even more critical. This paper will attempt to give the crucial parameters that affect this decision, from a technological, economical, and user perspective.

The paper is divided into the following sections:

I. Overview of Graphics from the Computer and Images from the Videodisc
II. Uses of Graphics Overlay
III. Videodisc Image Versus Computer Generated Pictures
IV. Segmented Images Between Disc and Computer
V. Summary

The use of these design guidelines are important in order to maximize the application objective versus the time/cost investment.

I. Overview of Graphics from the Computer and Images from the Videodisc

In most graphics overlay applications, a user views one screen. The participant or user of the application cannot generally detetermine from the viewing screen whether the current image is from the videodisc or from the computer. In addition, the most probable situation is that a given screen is composed of segments from the videodisc and the computer simultaneously. In a graphics overlay system, the designer has the option of overlaying computer generated graphics over the image generated by the videodisc.

In general, a microcomputer, equipped with a graphics adapter (similar to IBM's color graphics adapter), can generate text, text graphics, and pixel graphics. Assuming raster scan technology for the display device and a microcomputer equivalent to an IBM PC, PC/XT, or PC/AT, the system can generate lines, circles, rectangles, ellipses, polygons, etc., which constitute line drawings or filled areas on the screen. With adequate memory and sufficient microprocessor speed, three-dimensional figures can be produced and animated on the screen with varying degrees of shading and realism. The more realistic the picture, the more microprocessor speed and memory are required.

Videodisc images, on the other hand, are stored on a videodisc. This analogue information for a frame is embedded within the surface of the disc and retrieved through a laser or light reflecting mechanism. The image that is produced is a standard National Television Standards Committee (NTSC) signal with a fixed resolution of 640 by 483 pixels. The display screen used in the application should be of no higher resolution than that of the videodisc images. A still is produced by continually reading the same concentric circle (analogue information) from a CAV disc. Motion or animation is produced by displaying thirty frames per second from the videodisc source. To the user, sixty frames per second appears as motion. The videodisc motion sequence has the additional advantage of offering two different audio sources as the frames are shown from the disc. Unless the microcomputer has an audio card, the graphics images from the microcomputer do not have audio (except for sounds similar to musical notes produced by the internal microcomputer circuitry).

Overlay graphics, moreover, allows a user to view a motion sequence of videodisc images with the apparent insertion of computer generated graphics. To the viewer, the videodisc image and the graphics are combined. Unless still frame audio is available, a still videodisc image will not have audio whereas a motion sequence will.

II. Use of Overlay Graphics

The use of graphics overlay in a videodisc system will depend on the relevant application, but there are several general categories of usage. One obvious use of graphics overlay is the highlighting, marking, or emphasizing of details of images from the disc. As a motion sequence is running and the audio is describing object x, computer generated graphics can be used to underline or circle object x on the display screen. If an application presents a complicated videodisc image, graphics overlay can be used to mask out unwanted detail or unnecessary clutter. For example, if a videodisc image showing the internal circuitry of a microcomputer were being viewed, graphics overlay can be used to highlight or mask out various parts corresponding to the narration.

Another use of graphics overlay is the allowance of feedback or questions to the user of the system. Either from a still or a motion sequence, interactive user controls like stop, forward, or backwards can be imposed on the screen. In a training application, questions concerning key principles can be overlaid on corresponding videodisc diagrams. An additional use of overlay graphics is to change faulty or out-of-date information from the disc. In a commercial application using a videodisc as an informational database, overlay graphics can change the price of stock, advertise limited specials, or display relevant messages. In general, graphics overlay is the only convenient way to update the information on the videodisc. This is crucial in business applications that are likely to change in short time intervals.

A final use of graphics overlay is the simulation of motion along a path. Given a diagram, map, logical diagram, etc., lines produced by the computer can trace movement, giving further clarity to an otherwise static diagram. One example would be the tracing of routes between two destinations on a map, or the tracing of electron movement through logical circuit diagrams. Thus, in summary, graphics overlay techniques can be used to highlight, to emphasize, to mask out, to add information, to update facts, and to outline motion or movement on images.

III. Videodisc Images Versus Computer Generated Pictures

Which images orginate from which source is a critical design question in producing videodisc applications with graphics overlay. There are certain types of screens (frames) that are more usable and more cost effective for placement on the disc rather than creation from the computer. This section will attempt to give general design criteria for both sources.

For videodisc, the decision must be made early in the development cycle. Once the tape is produced for mastering, the videodisc images are permanent unless a new tape is made and another mastering is effected. Even though mastering one disc varies from one to two thousand dollars, the production cost of the graphics and personnel time are the skyrocketing dollar factors. Hence, it is obvious that upfront decisions on image placement will affect the efficacy and cost of the application.

To start the design criteria, an application that centers on interactive use of dialogue episodes are more appropriate for the disc. Training applications that simulate management dialogues, student teaching examples, and other human interaction scenes are the key images on a disc. Images that require extensive audio sections are more efficient from the disc than from the computer. Presently, it is more feasible to get five minutes of good audio from the videodisc rather than the microcomputer. Related to this category of senarios are those applications that are hard, expensive, or impossible to bring to the user or viewer in a true laboratory situation. Underwater exploring, pilot training, explosive demolitions, etc., would be areas that are suitable content for videodisc production.

In the areas of stills, the same design decision must be made. If the desired image is extremely detailed, intricate, or requires realism, it is probably better to represent it as a videodisc image rather than a computer generated graphic in real time. Of course, the still has to be produced somewhere and transferred to tape directly to disc (for a write once read many, WORM situation). The availability of existing slides, videotapes, etc., -- with legal permission to use -- may effect this design decision.

On the other side, certain categories of images are more efficient to produce as a computer generated image. The image may be stored as a memory dump or command file. The screen dump consists of character attributes or descriptive information about each pixel on the screen. The command file is a set of instructions which directs the graphics processor to draw on the screen the desired primitives (circle, line, etc.). Storage files of screen memory can be shown on the screen more quickly than the command files can produce an image; however, the command files are useful in updating segments or portions of the screen display.

Regardless of the method of storage, memory screen dump or command file, the graphic which is produced/displayed can be changed at anytime in the development process. The authoring system or programmer can change the screen video dump or the command file. This is not true for the video disc image. Image categories that fit the computer medium better are those stills which are simple block diagrams, such as a two level hierarchy chart or flow chart of three constructs. If the still is a diagram with simple rectangles and text, it can be easily generated from the computer. A designer may decide to include pictures of the actual central processing unit (CPU) on the videodisc, but may decide that a block diagram of the CPU should be a computer generated graphic.

Another image group that is better from the computer than the disc would be text screens. If the still consists entirely of text, the designer has more flexibility by generating this screen from the computer through the software rather than placing this frame on the videodisc. The text can be changed in color, size, length, and fonts at any stage of the development cycle. This is particularly important if the application is training and questioning is used as a branching mechanism or as a mastery tool. As the application is tested, the text screens can be changed if the wording is misspelled, vague, or ambiguous.

In the area of motion sequences, simple animation scenes can be generated from the computer. These scenes are probably more cartoon-like than realist in visual appearance. Short animated sequences which involve humor are better suited for computer generation. Since humor is relevant to the user and the application, comic episodes need to be changeable rather than permanent.

Thus in summary, motion sequences, intricate designs, and audio-based vignettes are more suitable videodisc graphics whereas simple logic diagrams, text screens, and short comic animation scenes are more suitable for computer generated images.

IV. Segmented Images Between Disc and Computer

With some broad distinctions between those images which should be designed for the disc and those that should be generated by the computer, an obvious conclusion is that a videodisc application can be more efficient and efficacious with the proper interplay between images from the two sources. This section will attempt to give design criteria for the efficient combining of the two image sources for an application. The assumption is made that the application objective is well defined, and that the guidelines of the previous sections have been followed.

In the application areas of vignettes or short narratives, overlay graphics can be used as transition mechanisms and summary devices. If a key point has been depicted by a short videodisc motion sequence, the designer can freeze a frame on the screen, and then produce a computer generated colored rectangle to mask off part of the screen (See Figure 1). The text appearing on the masked solid color portion could indicate the key point, interrogate the user, or serve as a branching mechanism.

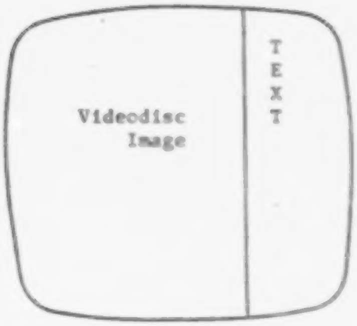


Figure 1: Text Screen over Portion of Videodisc Image

The video image could depict the realism of the concept presented whereas the simplicity of the text could reinforce the concept. In most cases, it is easier to read text against a non-busy colored background as opposed to text over an intricate video image itself. A light text color against a darker background color (white or yellow text on a blue or black background) is easier to read. When the application is being field tested, the textual displays over the still video frame can be changed or redesigned according to the user evaluation. This simple idea of text half screens on intricate images is quite useful in both transition and summaries with a still or motion sequences.

Utilizing graphics overlay, the designer, moreover, can place a control area on the screen during a motion sequence. This feedback area can be a rectangular area with prompts. Such controls as Repeat (the sequence), Stop (the Sequence) or Jump (to the end of this sequence. This interactive control panel can be solid or can appear as small areas with just one prompt within a small rectangle. (See Figure 2) The entire control panel is created via the overlay capabilities of the graphics system.

Another useful combination of the image sources is in an instructional application involving reading or using instrument with dials or gauges. An actual image of the instrument (microscope from the chemistry laboratory, cockpit from an airplane, or micrometer from a science class) can be placed on the videodisc. The graphics overlay system can overlay the labels on the instrument, identifying its parts, and the dials on any instruments. This use of overlay graphics produces a lot of needed exercises on reading instruments. To produce this situation requires only a few videodisc images and clever use of the line and text features of the overlay system.

Applications depicting complex motion can be divided between the computer and the disc. If the display shows a still or motion sequence from the videodisc, the overlay system can generate an animated pathway (Figure 3), similar to the appearance of marquee lights. This enforces the motion concept on the real environment. Applications depicting circuitry, laws of motion, sports, process control, etc., would be specific examples of this overlay classification.

Overlay situations, moreover, are useful in combing the real and the logical image of the same concept for the participant in the application. Using a still or motion sequences from the videodisc, the overlay subsystem can generate the logical diagram in varying formats. The screen could be split between the real picture on the right and the logical image on the left. Another format would be to impose a wire-like diagram over the realistic image from the

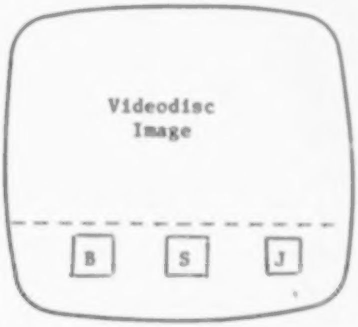


Figure 2: Control Panel

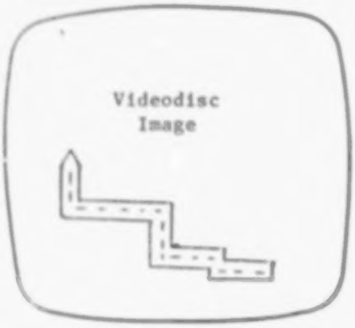


Figure 3: Animated Pathway Across Image

disc. Examples in this area include identifying or tracing the subsystems in an automobile engine, identifying the parts of a microcomputer, or communicating the flow of information in a corporation. Again combining or superimposing the proper segments from the computer on a disc frame is the key point.

V. Summary

In an application of a level III videodisc system with overlay graphics, the proper placement of images is crucial to the success of the application. Placing intricate stills, situation sequences with audio, and hard to produce images are more appropriate for videodisc placement whereas simple logic diagrams, block diagrams, and text frames are better suited for the graphics overlay unit. The most efficient usage of both images is the combining or overlaying of the two image sources on one screen. Using graphics overlay to facilitate transitions, to create control panels for the user, to superimpose different dials on instrumentation, and to indicate animated motion on complex diagrams are just a few broad categories of appropriate image design. The proper combining/superimposing of computer generated segments with videodisc images can enhance the success of the videodisc application.

References

Allen, D., (1986). When byte meets pixel. Videography, (pp. 76-80). October.

Christodowlakis, S., Faloutsos, C. (1986). Design and performance considerations for an optical disk-based, multimedia object server. Computer. December 1986, 45-53.

DeBloois, M.L. (1982). Principles for designing interactive videodisc instructional materials. In M.L. DeBloois (Ed.), Videodisc/microcomputer courseware design. Englewood Cliffs, NJ: Educational Technology Publications.

Donahue, T., & Donahue, M. (1983). Understanding interactive video. Training and Development Journal, 37 (12), 26-30.

Fujitani, L. (1984). Laser Optical Disk: The Coming Revolution in On-line Storage. Communications of the ACM. 27 (6), 546-954.

Grob, B. (1984). Basic Television and Video Systems. New York: McGraw Hill.

Hannafin, M., Garhart, C., Rieber, L., & Phillips, T. (1985). Keeping interactive video in perspective: Tentative guidelines and cautions in the design of interactive video. In E. E. Miller and M. L. Mosley (Eds.), Education media and technology yearbook (Vol. 11). Littleton, CO: Libraries Unlimited, Inc.

Hearn, D., Baker, M. (1986). Computer Graphics. Englewood Cliffs: Prentice-Hall.

Helgerson, L. Buying graphic overlay for videodiscs, EITV. October 1985, 52-58.

Lanier, R., The interactive videodisc is in motion. EITV. April 1986, 36-43.

Parsloe, E. (Ed.). (1983). Interactive video. Wilmslow, Cheshire, UK: Sigma Technical Press.

Park, S., (1985). Interactive microcomputer graphics. Reading, Massachusetts: Addison-Wesley.

Pigford, D. (1986). Graphics overlay for Interactive Videodisc Systems. Proceedings of the 28th International ADCIS Conference, p. 482.

Robinson, R. (1983). The Video Primer, pp. 311-320, New York: Putman.

Schwartz, M. D. (1982). Designing interactive video programs. In S. Floyd and B. Floyd (Eds.), Handbook of interactive video. White Plains, NY: Knowledge Industry Publications.

Sturm, R. (1985). High tech breakthrough: Interactive videodisc. Wilson Library Bulletin, 59, 450-52.

D.V. Pigford
Dept.of Computer Science
Western Illinois University
Macomb, IL 61455
(309) 298-1452

Statistics From The Top Down

Susan Athey - Oregon State University

Abstract

This paper describes a micro-computer based mentor system incorporating expert knowledge to lead students through the process of choosing a statistical technique to analyze a problem. The mentor uses a top-down approach based on the statistical question being asked, types of data, relationships between the data, and the number of variables. Results of tests showed that students choose the correct technique more often when using the mentor than when using a textbook.

Introduction

Undergraduate business curriculums require students to include statistics courses in their programs of study. These courses cover material in regression, ANOVA, hypothesis testing, and making inferences about a population based upon a sample. Included in the discussion of each technique are the assumptions and theory underlying each model, the steps to carry out the calculations, and the objectives of each technique. While enrolled in a statistics class, the students know which section or chapter of a book problems come from and use this as their frame of reference or anchor when choosing a technique. When they are faced, however, with the need to use statistics in some other setting, they no longer have this anchor and, consequently, have no idea what the appropriate technique is. Worse, students may have no idea how to go about deciding upon a technique. In other words, the student has acquired declarative knowledge (facts and formulas) but has acquired little or no procedural or conditional knowledge. They do not know how to go about selecting a technique or under which conditions a technique is appropriate.

Although numerous statistics packages (SPSS, SAS, MINITAB) exist which run on mini and micro computers to perform extensive calculations quickly and accurately, these packages do not offer any selection assistance to the user. Unfortunately, the analyst must determine for herself what analysis she wishes to perform and in what order the tests are to be performed. This is seen as a problem by persons both in and out of the education field (Gale and Pregibon 1985; Chambers 1981; Hahn 1985).

Now that computers are widely accepted for their computational skills in statistics, the next step should be to combine the micro-computer's power with rules incorporating procedural and conditional knowledge in order to help people decide how and when to apply statistical techniques. (HaKong and Hickman 1986; Remus and Kotteman 1986). This paper presents a mentor system developed for use by first or second statistics courses in a college of business. The mentor system is designed to assist a student in choosing a correct statistical technique. It is not designed to make the decision for the student.

System Description

In order to make this into a tractable problem, a subset of statistical techniques was chosen which consists of statistics commonly found in a business curriculum. Previously, authors have developed classification for relatively small subsets of statistics. Harris (1975) and Tabachnick and Fidell (1983) both developed classifications of multivariate statistics. Siegel (1956) developed a classification for non-parametric statistics. Other classification schemes include the extensive one by Andrews et. al. (1974), Morris and Rolph (1981), Anderberg (1973) and Walpole and Myers (1972). None of these, except for Andrews et. al., however, includes this particular set of business statistics.

The mentor is built around a knowledge base consisting of condition/action rules. These rules are obtained from a four dimensional table which clearly illustrates the particular combination of conditions needed to determine a technique. A table format was selected because it allowed the experts who validated the rules to make visual checks on the rules. If a cell of the table is empty, a technique is missing. If a cell contains more than one technique, additional conditions may be necessary to differentiate between the techniques.

The first dimension in the table consists of five statistical questions which these techniques can answer. The questions are:

1. What is the strength or degree of relationship, if any, between the variables?
2. Are there any differences between groups subjected to different treatments?
3. Can predictions of individual values or group membership be made?
4. Is there any pattern or structure to a large data set?
5. Are inferences about a population based on a sample needed?

The second dimension checks for a dependency relationship between the variables. The third and fourth dimensions determine data type (eg. quantitative, qualitative, ordinal, dichotomous) and number of variables, respectively. The table contains 30 different techniques and employs approximately 140 separate rules to distinguish between these techniques under different conditions.

A student using the mentor system is led through a series of screens by engaging in a dialogue with the mentor to describe the data being analyzed. If a student doesn't understand a question, HELP is available from screens of definitions and examples. The HELP is invoked using function keys. If the student decides the data have been described incorrectly, a CHANGE function is available . This function allows the student to add or delete variables or change the description of the variables.

Methods

The mentor was tested using two distinct student populations, 25 undergraduates enrolled in a second business statistics course at the University of Arizona and 19 MIS Ph.D. students (mostly first year) also at the University of Arizona. The undergraduates had each passed at least one micro-computer course and were offered the equivalent of two homework assignments credit to participate. They were also given the opportunity to earn $10. The $10 was a reward to the person in each of two assistance method groups who found the most correct solutions in the shortest amount of time.

Half of each sample was randomly assigned to a computer-assisted group and half to the textbook-assisted group. Each person in each group solved the same five statistics problems in the same order. The problems were standard problems taken from statistics textbooks except that all numbers were removed from them. The computer-assisted groups were divided into smaller groups of 2-4 in order to be more easily accommodated in the computer lab. These students were each assigned to an IBM PC and given directions for the use of the computer and the mentor software. They were directed to use the computer mentor to choose a statistical technique to be used to analyze each of the problems. They were not to do any calculations. Questions about the operation of the software and the hardware were allowed as the students worked. The textbook-assisted groups were also divided into small groups and each student had the same statistics textbook used in their class available as a reference. This group was also to determine the appropriate technique and could use their textbook as much as needed.

The time to complete each problem was noted as well as the technique chosen and the statistical question answered. Observations were collected during the sessions and verbal comments were solicited and noted at the end of each session.

Results

The data were analyzed for two characteristics: effectiveness of an assistance method as determined by the proportion of problems solved correctly and the average number of problems solved correctly; and efficiency of the methods as determined by the average amount of time required to solve the problems.

The undergraduate groups consisted of 12 students in the computer-assisted group and 13 in the textbook assisted group. The computer group chose 38 correct techniques (63%) while the textbook group had 27 correct solutions (41%). A chi-square test for a difference in the proportion of correct answers concluded that the computer group did solve a larger proportion of problems correctly. ($X^2 = 5.937$, $p=0.015$) The same test for the graduate students using 19 Ph.D. students also suggests that a difference does exist ($X_2 = 2.729$, $p=0.10$) but was not conclusive. [Table 1] An underlying reason that the findings were not conclusive for the graduate students is that several of them had just completed extensive data analysis work and, therefore, were quite familiar with different statistical techniques.

The average number of correct solutions per person was also measured. A t-test with

Table 1

Measuring the Proportion of Correct Answers

	Undergraduates		Ph.D. students	
	Textbook	Computer	Textbook	Computer
Number of Problems	65	60	45	50
Number Correct	27	38	26	37
Proportion Correct	41%	63%	58%	74%
Chi-square	5.937 (p=0.015)		2.729 (p=0.10)	

Table 2

Measuring the Average Number of Correct Answers

	Undergraduates		Ph.D. students	
	Textbook	Computer	Textbook	Computer
Number of Subjects	13	12	9	10
Number of Correct Problems	27	38	26	37
Average Correct	2.07	3.08	2.88	3.70
s^2	1.90	1.23	1.34	2.11
t-statistic	2.005 (p=0.028, one-tailed)		1.367 (p=0.09, one-tailed)	

Table 3

Measuring the Average Time to Solve Problems

	Undergraduates		Ph.D. students	
	Textbook	Computer	Textbook	Computer
Number of Problems	65	60	45	50
Average Time (minutes)	5.8	10.5	7.8	12.5
s^2	7.78	24.15	14.0	61.48
Times above median	16	47	17	30
Times below median	49	13	28	20
chi-square	36.0 ($p < 10^{-6}$) median = 7.0		4.678 ($p = 0.03$) median = 7.8	

the undergraduate group (t=2.005 with 23 d.f., p=0.028) suggests that the computer group did solve a larger average number of questions correctly than did the textbook group. Once again, the evidence is not as strong for the graduate students (t=1.367 with 17 d.f., p=0.09) [Table 2]

The textbook groups were observed as they worked through the problems. Most of the students searched through the textbook in no particular pattern trying to find an appropriate technique. After the experiment, the students explained that they were attempting to find an equation or technique to jog their memory. None of the students seemed to have a formal procedure or plan of attack to use to help them choose a technique. Even the Ph.D. students failed to use a methodical search procedure in their problem solving.

One should note that a large proportion of both undergraduate groups reached a correct solution for one of the problems. Questioning revealed that their statistics class was studying ANOVA and the particular problem was an application of ANOVA. The students used this as their frame of reference, allowing them to easily solve the problem. In fact, several students were so strongly anchored to the technique that they tried to solve even non-related problems with ANOVA.

A medians test was performed on the average time necessary to solve the problems for each assistance method. The results showed that a highly significant difference exists in the average time to solve the problems. The undergraduate computer-assisted group used an average of 10.5 minutes per problem while the textbook assisted group used only 5.8 minutes (p < 10-6). The difference was less dramatic but just as conclusive for the Ph.D. students (7.8 minutes vs. 12.5 minutes, p=0.03). [Table 3]

Several reasons exist for this time difference including the learning curve on a new system as well as on a new method of problem solving and the additional reading imposed by the computer system. The students could not just skim through the mentor as they did with the textbook. Increased usage of the system should decrease the time difference.

Discussion

The data and feedback from the students seem to indicate that an improvement in performance in solving statistics problems could be obtained through the use of the mentor system. Additional modules do need to be incorporated into the mentor, however. Observations and analysis of the reasons for the selection of incorrect techniques by the computer group indicated that students have difficulty determining what statistical question they are actually trying to answer about the data. A module to assist with this phase of the analysis is needed, perhaps based on key words or phrases in problems. Additional modules are needed to emphasize and reinforce the assumptions which underlie the techniques and which are key to making correct statistical decisions. This module could consist of tests for each assumption and additional rules to be invoked if the assumptions are false. These additional rules would lead the student to other statistical techniques with less stringent assumptions.

The next step in this work will be to incorporate this mentor into an actual classroom and build an entire course around this upside down methodology. Given the many problems students have learning statistics, the time has come to integrate computer tools and logic and find a better way to teach and learn statistics. Prolonged use of the tool should allow the identification of additional problem areas in the selection of techniques and will allow the mentor to be upgraded and improved-all with the aim of helping students to able to use statistics after they leave the classroom.

REFERENCES

Anderberg, Micheal R. (1973) Cluster Analysis for Applications. New York: Academic Press.

Andrews, Franklin; Klein, Laura; Davidson, Terrence N.; O'Malley, Patrick M.; and Rodgers, Willard L. (1974) "A Guide for Selecting Statistical Techniques for Analyzing Social Science Data". Ann Arbor, MI: Institute for Social Research.

Chambers, John M. (1981) "Some Thoughts on Expert Software". Proceedings of the Computer Science and Statistics: 13th Annual Symposium on the Interface, edited by William F. Eddy. pp. 36-45.

Gale, William A. and Pregibon, Daryl. (1982) "An Expert System for Regression Analysis". Proceedings of the 14th Annual Symposium on the Interface of Computer Science and Statistics. New York: Springer-Verlag, pp. 110-117.

Hahn, Gerald J. (1985) "More Intelligent Statistical Software and Statistical Expert Systems: Future Directions". The American Statistician. February 1985, pp. 1-8.

HaKong, L. and Hickman F.R. (1986) "Expert Systems Techniques: An Application in Statistics". Proceedings of the Fifth Technical Conference of the British Computer Society Specialist Group on Expert Systems, edited by Martin Merry. Cambridge, Great Britain: Cambridge University Press.

Harris, Richard J. (1975) A Primer for Multivariate Analysis. New York: Academic Press.

Morris, Carl N. and Rolph, John E. (1981) Introduction to Data Analysis and Statistical Inference. Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Remus, William E. and Kotteman, Jeffrey E. (1986) "Toward Intelligent Decision Support Systems: An Artificially Intelligent Statistician". MIS Quarterly, December 1986. pp. 403-418.

Seigel, Sidney. (1956) Non-parametric Statistics for the Behavioral Sciences. New York: McGraw-Hill.

Tabachnick, Barbara and Fidell, Linda S. (1983) Using Multivariate Statistics. New York: Harper and Row.

Walpole, Ronald E. and Myers, Raymond H. (1972) Probablility and Statistics for Engineers and Scientists. New York: The MacMillan Co.

Dr. Susan Athey
College of Business
Oregon State University
Corvallis, Oregon 97331
503-754-3687

AUTHORING SYSTEM EVALUATION PACKAGE

EMILE ATTALA AND MARSHA EPSTEIN

CALIFORNIA POLYTECHNIC STATE UNIVERSITY

ABSTRACT

In CAI, as in any area of Computer Science, the choice of computer language may be critical to the success of a project. In the example presented in this paper, an attempt was made to use Apple PILOT to develop a fairly substantial medical application. There were limitations which could not be overcome. This experience led to the development of an Authoring Systems Evaluation Package. It allows the most important features of an authoring system to be isolated and weighted according to the user's needs. This scheme could be used to evaluate any free-format language as an authoring tool.

Historically, CAI has been relegated to the classroom for use in the mundane drill-and-practice or tutorial modes of instruction, but it has far greater potential. With imagination, CAI can be extended into less-traditional areas. More and more, CAI systems are found in training and rehabilitation centers. Therefore, when it was decided to use the computer to treat stuttering, one purpose was to demonstrate the versatility of the computer, and by implication, CAI.

The choice of a computer language may be critical to the success of any application. As the programmer goes from one application to another, the demands change. In the example presented here, real-time graphics were an integral part of the program. Our attempt to use Apple PILOT met with failure. This indicated to us that there was a need to evaluate an authoring system before money and time were wasted on an inadequate package. A consistent evaluation scheme which allowed the user to compare authoring systems was needed.

The Problem. Stuttering is the inability by a speaker to coordinate respiration, phonation, and articulation. The result is the familiar repetitious, halting speech which has shown itself remarkably resistant to therapy. Speech pathologists have had little success treating such patients with the traditional one-on-one office visit. Auditory feedback from a therapist has not provided good results.

The idea for a computer program to treat stuttering came from a speech pathologist who believed that visual feedback of respiration and vocalization would allow patients to spend hours in front of the computer, working to improve their speech, in a relaxed, non-threatening environment.

It was important that the program fulfill certain requirements:

1. The feedback system must measure the patient's respiration. Many stutterers have difficulty coordinating respiration with vocalization. If the patient's respiration could be instantly displayed graphically on the computer screen in real time, then the stutterer could work to control his respiration while getting visual feedback from the computer.
2. The system must measure the patient's vocalization, i.e., a visual record of when he makes sounds and when he does not make sounds. Since the problem is one of coordination, the element of phonation must also be displayed graphically in real time so that the stutterer can work to coordinate

both vocalization and respiration.

3. There must be a facility which directs the stutterer to utter certain words or phrases. Stutterers often have difficulty with particular sounds, syllables or words. The therapist should have the option of entering a series of words or phrases for the patient to use in his therapy. The patient would be prompted to say these words/phrases, which he would practice while at the same time monitoring his respiration and vocalization.

4. There must be a mechanism to put pressure on the patient while he is attempting to speak. Stutterers often have difficulty speaking in a pressured situation. Therefore, there should be an element of time constraints so that the patient is actively pushed to vocalize as quickly as possible. There should be images on the screen by which he can determine if his effort met with success or failure.

5. This therapist already had in place an Apple IIe Computer with dual floppy disk drives and a color monitor. Therefore, it was necessary for the program to be compatible with that computer.

still require a fair level of programming skill. To eliminate the need for coding, authoring systems have been developed. An authoring system allows a non-programmer to harness the power of the authoring language. An authoring system is completely menu-driven. There is no coding; no need to enlist the support of a professional programmer.

Authoring languages have been used to write programs for use in rehabilitation medicine. These include vocabulary programs for aphasics, language programs for children, and sign language programs for both those who are hearing impaired and those learning sign language as educators. The National Institute for the Deaf has interfaced a microcomputer program with color television taped sequences. It is used to display sign language or speech reading (lip reading) along with the standard computer screen text. These programs harnessed the power of authoring languages for use in a non-traditional format.

An authoring language, Apple PILOT, was the initial language of choice for the development of this project because it is available on the Apple II machine.

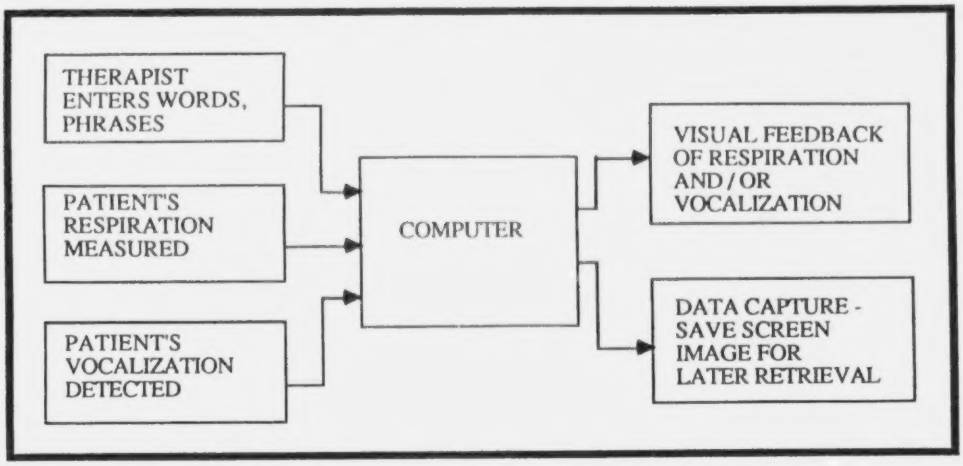

THERAPIST ENTERS WORDS, PHRASES
PATIENT'S RESPIRATION MEASURED
PATIENT'S VOCALIZATION DETECTED
COMPUTER
VISUAL FEEDBACK OF RESPIRATION AND / OR VOCALIZATION
DATA CAPTURE - SAVE SCREEN IMAGE FOR LATER RETRIEVAL

Figure 1 - System Requirements for Stuttering Therapy Project

Finding a Solution. Because this project was intended as a demonstration of the versatility of CAI, it was considered desirable to use a CAI-oriented language to develop the system. While any computer language can be used for CAI, special languages - called authoring languages - have been developed to facilitate the computerization of lessons, record-keeping, testing and grading procedures. Authoring languages are high level computer languages, but they

Apple PILOT, an early version of the PILOT language, is designed for an 8-bit machine and lacks the speed and versatility of the PC PILOT, which runs on a 16-bit microcomputer. Apple PILOT proved inadequate for the task because of just one serious fault. The major requirement of this project was that of nearly-instantaneous feedback to the patient. When the patient vocalizes, the drawing on the screen must register that change immediately. Apple PILOT's graphics turned out to be hopelessly slow. The language has the capability of loading entire pictures from disk to the screen very rapidly, but real-time graphics were a different matter. If the project were to continue, another language had to be found.

The decision was made to work in Applesoft BASIC. This language was chosen because it is native to the APPLE II - it exists in ROM. Therefore there is no problem with licensing or portability. Furthermore, BASIC may be considered a CAI language. It has rich graphics capabilities, it can manipulate files, read the game port, the keyboard, or whatever input device is desired. It was not necessary to prepare user-written assembly language routines in this project. This is because breathing and vocalization are slow processes and the game port could sample the data at a great enough rate in a BASIC program. A fast process like a nerve impulse (a couple of milliseconds) would require assembly language control of the paddle port. The Apple computer has built-in assembly routines to poll the keyboard and reset error traps. These routines were accessed and are listed in the "Applesoft BASIC Programming Reference Manual."

It is common to find that one language or one system is not well-suited to the user's needs. Before a potential user makes the substantial purchase of computer hardware or a language system, there should be a systematic way for the user to evaluate his needs to avoid costly mistakes. Apple PILOT was found to be unsuitable for the development of this project because of just one serious deficit that could not be overcome. In the same manner, other users have undoubtedly run into similar problems. For example, a system may prove difficult to learn or use; the documentation may be inaccurate or unclear; the system may not permit graphics and text to exist simultaneously on the screen; it may lack subscripted variables, particularly string variables. Any of these examples could prove to be a serious weakness in a system. Therefore, in conjunction with the Stuttering Therapy Project, an Authoring System Evaluation Package was developed.

Authoring Language Evaluation System.

The Authoring System Evaluation Package was designed to inform a non-technical person of the strengths and possible pitfalls of authoring systems. It permits the potential user to make a reasoned judgment as to which system best suits his needs. At first, all systems look exciting and versatile when being demonstrated by a salesperson. The brochures and demonstration programs always show dazzling graphics and glossy lessons that more closely resemble computer games than educational tools. Can an ordinary user make a reasoned judgment in the face of such sophisticated marvels? Do the systems do what is required easily and reliably?

The Authoring System Evaluation Package provides a method for comparing and evaluating authoring systems. The characteristics that make these systems most useful are determined: their strengths and weaknesses. A weighting factor was developed for each trait so that a user considering purchase of a system can arrive at a weighted score for each package under consideration. A comparison of these scores provides an objective method of choosing among different systems.

When the decision is made to purchase an authoring system, a choice must be made as to which system best fills the user's needs. Some elements of this decision are easy to make. If a computer is already on site, then the system purchased must run on that computer. (Such was the case with the Apple II and the stuttering project.) If an entire package -- both computer and software are to be purchased -- then there are many more options available. Price is another element which may quickly eliminate some systems.

Once the field has narrowed somewhat, how should the final decision be made? How should each system be evaluated? Each system is loaded with features. Which features are useful and should enter into the decision-making process and which are less important and can be safely ignored? A faulty decision could be costly.

This Authoring System Evaluation Package was designed to take the guesswork out of the decision-making process. It presents a list of features, each explained at some length (Figure 2). Alongside each feature is a series of asterisks -- the weighting factor. The weighting factor represents the relative value given to each feature. The most critical items are given the maximum weight (five asterisks) and the least important items have only one asterisk. All the rest reside somewhere between the two. A feature which carries the greatest weight should be implemented on any authoring system under consideration because the lack of this feature would so weaken the system as to render it

ACCESSIBILITY

SECTION 1

	ITEM	WEIGHTING FACTOR	GRADE					SCORE
			A	B	C	D	F	
1.	System is easy to learn and to use	*****						
2.	Manuals are easy to read, complete, and accurate	*****						
3.	Lesson can be run in author mode	**						
4.	Reliability of the system	***						
5.	Number of instructions available	**						
6.	Help screens available as needed	***						
7.	Understandable error messages	***						

Sum of items #1 through #7: _____
This is the score for Section 1.

Figure 2 - The evaluation sheet for Section 1: Accessibility

AUTHORING SYSTEM EVALUATION

SYSTEM NAME: ______________________________

VENDOR NAME: ______________________________

SECTION 1 - ACCESSIBILITY SCORE________
SECTION 2 - RECORD-KEEPING FACILITIES SCORE________
SECTION 3 - SYSTEM CONFIGURATION SCORE________
SECTION 4 - VENDOR ACCESSIBILITY SCORE________
SECTION 5 - USE OF TEXT IN LESSONS SCORE________
SECTION 6 - GRAPHICS CAPABILITY SCORE________
SECTION 7 - PROGRAMMING POWER SCORE________
SECTION 8 - ADDITIONAL FEATURES SCORE________
SECTION 9 - COST & COPYRIGHT AGREEMENTS SCORE________

TOTAL SCORE______________

TOTAL POSSIBLE SCORE IS 788

$$\frac{\text{TOTAL SCORE}}{\text{TOTAL POSSIBLE SCORE}} = \frac{\_\_\_\_\_\_}{788} = \_\_\_\_\_\_ \%$$

Figure 3 - The Authoring System Evaluation Package Score Sheet.

unacceptable.

In addition to the weighting factor, there is a grading scheme (A through F) to be used by the evaluator. As the evaluator (perhaps a potential user of the system) proceeds, he/she should examine each feature as it is presented on the system and give it a letter grade. This is accomplished by simply checking the corresponding box. A score for each item is computed using the traditional 4 points for an A, 3 points for a B, 2 points for a C, 1 point for a D, and 0 points for a grade of F. The weighting factor, multiplied by the grade points, yields the score for that item. Therefore, the maximum score for any item is 20 points (5 asterisks * 4 points) and the lowest possible score is zero. A score of zero does not necessarily mean that a feature is entirely absent. It could also mean that it is so poorly implemented as to be nearly useless.

The authoring language features are grouped into nine categories, each category on a separate page. At the bottom of the page is a space in which to enter the sum of the scores for all items in that category. This sum is also entered on the cover sheet of the evaluation package. After the evaluation process has been completed and all scores computed, a total score (the sum of the scores for all nine categories) is calculated (Figure 3). This total score is divided by the total possible score and a percentage is the result.

If several authoring systems are evaluated in this manner, each will have a numeric score: the percentage which is the ratio of total score to total possible score. Since the systems were evaluated using the same criteria, these percentages may be compared with one another. One can visualize a set of scores ranging, perhaps, from 49% to 85%, with preference given to the system with the highest score.

How should the evaluator proceed with the evaluation process? The ideal situation would be for the potential author to sit down with a copy of the authoring system, armed with a complete set of documentation, and test each feature deemed important to the author's requirements. This is probably an unrealistic proposal. Many potential authors are not equipped with the time or expertise necessary to perform such a task. Perhaps the best that can be expected is that the author-to-be will sit at a machine on which the system is running and get a feeling for the appearance of the system, some of its commands; how to move around from the text editor to the graphics editor, from author mode to student mode, and back; how quickly the system operates; the resolution and color of the graphics; the appearance of the text; etc. Perhaps the potential author will watch an experienced person take the system through its paces and see some artfully-prepared demonstration lessons.

One suggestion is that it may be very useful for both the author and the vendor to fill out the Authoring System Evaluation form. The potential user knows his needs and the needs of his associates who may also be using the system. He can use the manuals and other documentation to assess the options available but he cannot be as familiar with the system as is the vendor. If a technical representative of the vendor were to also fill out this form, perhaps with the user alongside, and provide explanation along the way, this would greatly clarify the system for the user. The user could then compare his own responses with those of the vendor, to see where they diverge. If it were to be used in this manner, this package could be both an evaluation tool and a teaching tool.

REFERENCES

Apple Computer Inc. Applesoft II Basic Programming Reference Manual. Apple Computer Inc. 1978.

Epstein, Marsha M. Microcomputer Therapy for Stutterers with a Special Emphasis on the Evaluation of Authoring Systems. Master's Thesis, California Polytechnic State University, 1986.

Schwartz, Arthur H. Handbook of Microcomputer Applications in Communication Disorders. College-Hill Press, Inc., 1984.

Van Riper, Charles. The Nature of Stuttering. New Jersey: Prentice-Hall, Inc. 1971.

Wagner, William J. "Author Languages: Instruction Without Programming," Classroom Computer News. Vol. 1, No. 6. July-August, 1981.

Marsha Epstein
1764 Jami Lee Ct.
San Luis Obispo, CA 93401
(805) 543-9352

AN EVALUATION OF THE TEACHING EFFECTIVENESS OF A PC-BASED EMULATION PROGRAM FOR PRODUCTION COURSES

K. KERN KWONG, CALIFORNIA STATE UNI., L.A.

C. HIRANYAVASIT, S. MOON, GEORGIA STATE U.

ABSTRACT

The presentation discusses the project undertaken by an institute to design and develop an emulated program of the IBM's MAPICS (manufacturing accounting and production information control system). MAPICS is a comprehensive industrial information system which covers topics such as production, accounting, financial analysis, and sales/distribution. MAPICS runs on IBM minicomputers, and is too expensive for a university to operate such system for pedagogical purposes. The project's products include: a set of four diskettes, an instructor's guide, a nine-units student workbook. The program covers applications such as data base, production planning and scheduling, inventory management, material planning, order management, and production control. The emulated program was developed by capturing the selected screens from MAPICS, and by linking these screens together to emulate the real information system. The emulated program runs on IBM PC with 256K of memory. Basically, this PC-based program displays the same screen that students would see if they were using a real MAPICS. Such emulation program has several advantages over the real system. For example, easy to use by inexperienced students, tailored help screens, inexpensive to operate by institutes, program and courseware are very portable. Currently, the program is being used, in several institutes, to teach production management courses in both graduate and undergraduate levels.

INTRODUCTION

As computers become prevalent in industry, the need for students to obtain a better understanding of computer-based manufacturing information systems becomes an important issue. In order to keep pace with this industrial trend, most recently-published Production and Operations Management (P/OM) textbooks allocate a considerable portion of the text to the concepts of integrated manufacturing information systems. Furthermore, many universities are developing a variety of instructional tools to facilitate students' understanding of manufacturing information systems in introductory and advanced P/OM courses (Chong, 1985; LaForge, 1986; Schuette, 1984, 1986). A common, but expensive approach, is to install a real industrial manufacturing information system for use in these courses. One university, for example, installed two IBM System/36s and developed a series of laboratory exercises to teach students of IBM's industrial manufacturing information system, the Manufacturing Accounting and Production Information Control System (MAPICS).

Although the use of a commerically available industrial information system is probably the best way to invoke a high level of student interest and motivation, it also generates some practical problems, the major ones being system complexity and high cost. Since industrial manufacturing information systems were originally designed for industrial use, they are mostly too complex for inexperienced students and instructors. In addition, the cost of purchasing and maintaining an industrial manufacturing information system (including hardware, software and personnel) is too excessive for most universities.

To resolve these problems, a Personal Computer-based (PC-based) emulation program was developed as an alternative to the real industrial system. Basically, this PC-based program displays the same screen that students would see if they were using a real industrial manufacturing information system. The emulation program was developed by capturing the selected screens from the real system, and by linking these screens by a series of IBM BASIC programs. The emulated program runs on IBM and IBM-compatible Personal Computers with 256K of free memory. With a limited data base and without dynamic data processing capability, this PC-based emulation program provides the same kind of information and "hands-on" experience that the corresponding industrial system does. It is important, however, to ensure that the PC-based program is as effective as the System/36-based exercises.

PURPOSE OF THE STUDY

If the PC-based emulation program is as pedagogically effectively as a real industrial information system, it will provide significant advantages. Since most universities already have their own PC laboratory, the use of the PC-based emulation program will not involve much additional expense. Furthermore, the PC-based emulation program can be optimized for educational effectiveness. Since industrial manufacturing information systems include many details not essential for the understanding of basic concepts, the PC-based emulation program can be made much simpler. In this regard, the purpose of this study is to compare the effectiveness of these two educational approaches: using the actual industrial information system and using a PC-based emulation program.

CRITERIA

The two educational approaches were compared on three criteria: students' mastery of the subject matter, ease of using the media, and students' attitude toward the media. In this experiment, the students were randomly divided into two groups. One group was trained by using the real manufacturing information system; the other, by the PC-based emulation program. Both groups were required to work on computer exercise materials which have been especially designed to help them utilize the information provided by the manufacturing information system in making managerial decisions. These exercises were identical for both groups, except for operating instructions of the terminal or the personal computer.

To measure of students mastery, pre-test and post-test were used to evaluate the students' knowledge of the basic concepts of a manufacturing information system. To measure ease of use, the number of student's requests for assistance and the time each student spent in the exercises were recorded. To measure the students' attitude, a survey questionnaire utilizing a Likert scale was administered at the end of the post-test. This attitude questionnaire gave such information as the students' perceived learning, the ease of using the media, and the students' opinion of the overall relevance of the computer exercise.

EXPERIMENTAL DESIGN

Three undergraduate and one graduate class of the introductory P/OM course participated in this experiment. Each class was randomly divided into two groups with one group assigned to the industrial system and the other to the PC-based emulation program. The experiment had basically two sessions. Each session took one class period of approximately two hours.

In the first session, a standardized lecture on the basic concepts of a generic manufacturing information system was given to each class by the same instructor in order to reduce the instructor bias. At the end of the lecture session, a pre-test was given. In session two, students were assigned to use either a personal computer or a terminal on the System/36 to work on the computer exercises. While the students were working on the computer exercise, the number of students' requests for assistance on the exercise were recorded. At the end of the computer exercise, the students were given a post-test and an attitudinal questionnaire. In summary, the experimental design used in this research was a two-treatment pre- and post-test design.

RESULTS

The test results of a t-test analysis are discussed in the following paragraphs. There was no significant difference between the means of the two groups in their pre-test and post-test scores (alpha = 0.05). In other words, the null hypothesis that there was no significant difference, in terms of educational effectiveness, between the emulated program and the real manufacturing information system could not rejected. There was a significant difference, however, between the two groups with respect to the number of assistance requested by students during the computer exercises (alpha = 0.05). Students using the real manufacturing information system requested more help from instructors. This was probably due to the complexity of using the real manufacturing information system as compared to the emulation program, and the use of IBM PCs as emulated terminals for the System/36s in the MAPICS laboratory. Though students on the emulation system spent less time on the computer exercise than their counterparts on the real manufacturing information system, the null hypothesis could not be rejected at the 0.05 level. Finally, the responses to the six attitude questions were not significantly different between the two groups at the 0.05 level.

IMPLICATIONS AND CONCLUSIONS

This study indicates that the pedagogical effectiveness of the PC-based emulated program is not significantly differed from the real manufacturing information system. Thus, a PC-based emulated program can be used to teach students the concepts of complex manufacturing information system, without paying the cost of installation and maintaining a real system. In terms of transportability, an emulated program developed in one instituion can be readily adopted and used in other institutions.

The emulated program used in this study is part of a PC package consisting of three set of materials: a two-diskette PC-based emulation program for the MAPICS system, an eight-unit student laboratory workbook, and an instructor manual that provides the necessary training and basic operation guidelines for an instructor. Currently, the package is being tested and used by several universities in the United States.

With a PC-based emulation program, students can be taught advanced manufacturing information systems in P/OM courses at modest costs. This study indicates that a PC-based emulation system can be as effective, for educational purposes, as the real manufacturing information system.

REFERENCES

1. Chong, P.S., "Manufacturing Resource Planning Games," Production and Inventory Management Review, August 1985, pp. 26 and 56.
2. LaForge, R.L., & McNichols, C.W., "An Integrative, Experimental Approach to Teaching Production Management", finalist of the 1986 Instructional Innovation Award Competition of the Decision Sciences Institute, November 1986, Hawaii.
3. Schuette, H.L., "Educating Executives: Computers Bring The Factory to The Classroom," Management Review, March 1984, pp. 15-21.
4. Schuette, H.L., "Experience with Realistic Manufacturing Software in an MBA Operations Management Laboratory", paper presented at the Conference of The Institute of Management Sciences, April 1986, Los Angeles.

Kern Kwong
Department of Management
California State University-LA
Los Angeles, CA 90032
(213) 224-2961

SIMULATING AN ARMY AVIATOR QUALIFICATION COURSE:

RESOURCE ALLOCATION MODEL

Carol A.W. Sprunger
Universal Energy Systems, Inc.
Aviation Training Office

Paul J. Tremont
U.S. Army Research Institute for Behavioral and Social Sciences
Aviation R&D Activity

ABSTRACT

A model of an Aviator Qualification Course (AQC) was built to examine resource allocation during the design phase of Army courses by developing a discrete event model for simulating varying course configurations and conditions. The model consists of a network representing all AQC activities. Within this network each discrete course event was assigned an event name, a duration (with mean and standard deviation), and a probability of occurrence. The model was validated by initializing it to simulate known conditions, then comparing its output with output obtained under the same conditions in six AQC classes over a five month period. For each experimental simulation, the model was configured to reflect a course Program of Instruction (POI) of interest and initialized to represent resource allocation decisions in accordance with the specific purposes of the simulation. Results showed that simulations such as these offer a method for examining the effects of various course designs on resource consumption prior to implementation.

INTRODUCTION

All initial entry and aviator qualification courses (AQCs) are trained at the United States Army Aviation Center (USAAVNC), Fort Rucker, Alabama. The expendable and nonexpendable resources required to support these courses are extensive. A method to observe the effects of proposed design changes on resource consumption in these courses is needed.

Resource Allocation Questions

In order to examine resource allocation in the AQCs, training designers and developers will need to answer the following kinds of questions:

> What are the usage and availability levels of resources (e.g., IPs and aircraft) at any given time?
>
> How would altering the availability of selected resources affect various aspects of the course?
>
> What effect would changing the proportion of flight simulator time to actual flight time have on resource consumption?

Alternate Approaches to Examining Resource Allocation

Several approaches to providing answers to the above questions are possible. Comparisons of relative resource consumption and student throughput in each AQC was one method considered. Another approach would be to change a course and observe the resource consumption outcomes. A close analysis revealed serious limitations to each alternative. A simulation approach was therefore chosen for examining AQC modifications. Simulation provides three principle advantages: (1) it permits time compression, (2) it permits controlled observation, and (3) it does not disturb the functioning system, (Adkins and Pooch, 1977).

Approach to Model Design

Of the types of simulation models in common use, most can be classified along the following dimensions: static versus dynamic, continuous versus discrete, and deterministic versus stochastic, (Graybeal and Pooch, 1980). The uses of the model and the characteristics of the system to be simulated determine the type of model to be employed. The uses of the AQC model have been considered; below is a description of the characteristics of the AQC system to be modeled.

The AQCs are dynamic systems. Three classes in the 30-plus day training cycle simultaneously move through a different segment of approximately 330 events comprising these courses. As training is delivered, more than 150 variables change, including the number of aircraft in use, scheduled flight hours in the aircraft, and makeup flight hours. A dynamic computer model was therefore required to simulate these changing events.

The functioning of these AQCs is best describable in discrete terms. All activities within each course move through a series of separate system states in a discontinuous manner. Because the transitions between these system states are not gradual, the design of the AQC model was digital rather than analog, i.e., it used discrete rather than continuous transitions. Moreover, as Garzia and Garzia (1986) have observed, with resource management problems, discrete event methodology closely matches the way the systems being studied really work.

Another characteristic of these AQCs to be simulated was the uncertainty associated with various critical events which occur during training. These uncertainties are (1) the frequency of occurrence of inclement weather, (2) the frequency of occurrence of unsatisfactory evaluation flights, and (3) the duration of tasks within each flight period. During a typical one week period, three classes in residence in each AQC could encounter a total of approximately 90 opportunities for weather, evaluation flight outcomes, and flight period duration uncertainties. The connection between these uncertainties and resource consumption led to the stochastic design of this model. The assumption was made that these uncertainties are normally distributed.

The AQC model presented herein is, thus, a dynamic, discrete, stochastic model. The model consists of a computer program which, when executed, simulates the AQC by advancing through each course event while generating probabilities appropriate to the activities within each event and concurrently calculating resource expenditures.

METHOD

Design Model

Review documentation and interview subject matter experts (SMEs). A review and analysis of the current Program of Instruction (POI), an AQC Flight Training Guide, and a Master Scheduling Sheet was performed, supplemented by Army personnel interviews.

Analyze an existing AQC configuration and design an alternate AQC configuration. One version of the model was designed to track an AQC event sequence as specified in the Master Scheduling Sheet and as is currently being implemented. In a second version, flight simulator time was increased by 1.5 hours per student and aircraft flight time was decreased by 1.5 hours per student. To permit this change, time not currently used for academic instruction or flight training was redirected for use as flight simulator time. In addition, the alternate configuration requires two flight simulators whereas the existing configuration requires only one. The 32 day course length remained unchanged for both configurations.

Locate each AQC event for each configuration in a flowchart. A flowchart was designed for each course configuration to specify the daily event sequence of a typical class. The flowchart consisted of two networks: a main network and a subnetwork. A portion of the main network for a selected AQC, Training Day 17, is depicted in Figure 1. The main network contains all administrative and academic activities, as well as time spent outside AQC training periods.

The subnetwork, depicted in Figure 2 and entitled the "Flight Network (FN)," contains the six tasks comprising each flight period. This sequence of events occurs 40 times throughout the selected AQC. (Notice that a weather designation is made after the "Pre-Flight Aircraft" event.) The main network branches to this subnetwork for each flight period and subsequently returns to the main network.

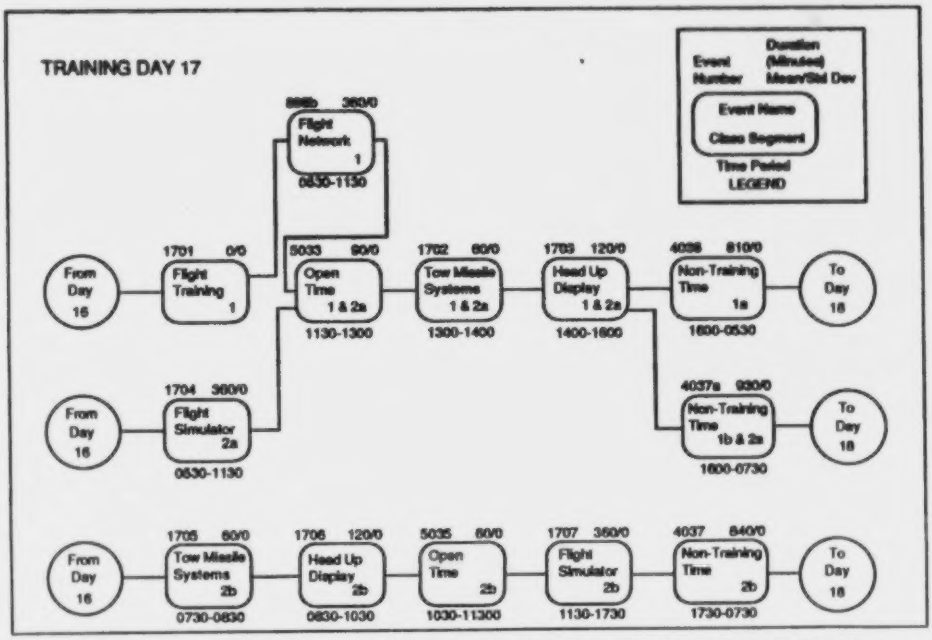

TRAINING DAY 17
Event Number
Duration (Minutes) Mean/Std Dev
Event Name
Class Segment
Time Period
LEGEND
888b 360/0
Flight Network 1
0830-1130
1701 0/0
Flight Training 1
5033 90/0
Open Time 1 & 2a
1130-1300
1702 60/0
Tow Missile Systems 1 & 2a
1300-1400
1703 120/0
Head Up Display 1 & 2a
1400-1600
4036 810/0
Non-Training Time 1a
1600-0530
From Day 16
To Day 18
1704 360/0
Flight Simulator 2a
0830-1130
4037a 930/0
Non-Training Time 1b & 2a
1600-0730
1705 60/0
Tow Missile Systems 2b
0730-0830
1706 120/0
Head Up Display 2b
0830-1030
5035 60/0
Open Time 2b
1030-11300
1707 360/0
Flight Simulator 2b
1130-1730
4037 840/0
Non-Training Time 2b
1730-0730

Figure 1. A portion of the main network depicting activities of a typical training day (Training Day 17).

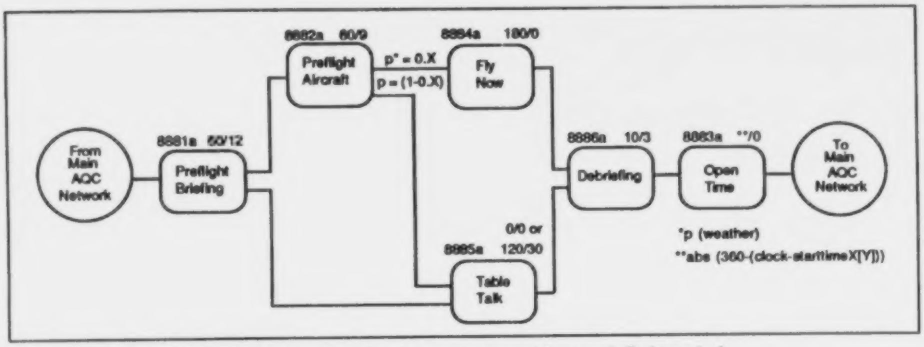

From Main AQC Network
8881a 50/12
Preflight Briefing
8882a 60/9
Preflight Aircraft
p* = 0.X
p = (1-0.X)
8884a 180/0
Fly Now
0/0 or
8885a 120/30
Table Talk
8886a 10/3
Debriefing
8883a **/0
Open Time
To Main AQC Network
*p (weather)
**abs (360-(clock-starttimeX[Y]))

Figure 2. The "Flight Network" depicting the six events comprising each flight period.

Build Model

Program the event sequence within an AQC class. Each event within a class was entered into the simulation program and assigned an event number, a mean time in minutes for its completion (according to POI specifications), and a standard deviation time (according to SME interviews). Micro SAINT, a product of Micro Analysis and Design of Boulder, Colorado, and IBM-compatible microcomputers were used for the development and execution of the model.

Program the event sequence among AQC classes within the selected course. The relative event sequence among AQC classes was programmed into the model as depicted in Figure 3. New classes began at ten day intervals. Observe that a minimum of three and a maximum of four classes receive training simultaneously.

The training of six complete classes was simulated in this study. However, only the activities of Class 3 were examined because it alone competed for resources with classes both ahead and behind itself throughout its entire duration.

```
                           Training Days
          /-----------------------------------------------\
Class 1   21--22...30--31
Class 2   11--12...20--21--22...30--31
Class 3    1---2...10--11--12...20--21--22...30--31
Class 4                 1---2...10--11--12...20--21--22...
Class 5                              1---2...10--11--12...
Class 6                                           1---2...
```

Figure 3. Event sequence among an Aviator Qualification Course (AQC) classes.

Incorporate calculations appropriate to each event. There were three types of calculations associated with the individual events of this AQC model: calculations to increase or decrease the levels of resources such as aircraft and IPs associated with an event, calculations to accumulate flight training time during an event, and calculations concerning the effects of inclement weather and unsatisfactory evaluation flights.

Design output files. Files were designed to capture the values of selected variables at specific times during model execution. The records contained in these files permitted off-line examination of the simulation outcomes.

Validate Model

Data on six AQC classes (within a selected course) was extracted from an Aviation Training Battalion's historical records. These records had been placed in a database as part of a separate effort, (Coffield, 1987). Of interest were the mean student/IP ratio, students per class, and IPs and aircraft available for each flight period. The frequency of inclement weather was also extracted from this database. Subject matter experts were interviewed and a second database was referenced to determine the frequencies of unsatisfactory evaluation flights and their repercussions.

Exercise Model

Select configuration, conditions (independent variables), and dependent variables. The version of the model appropriate to the purpose of the simulation was selected. The version choices were either the existing AQC configuration or an alternate configuration. After selecting the course configuration to simulate, the conditions of each simulation were specified by initializing the independent variables. These conditions were also determined by the purpose of the specific simulation. The dependent variables were associated with the outcome of each simulation.

Run simulations. Three simulations were performed: a Historical Simulation, a POI Baseline Simulation, and an Experimental Simulation. The configuration, conditions, and purpose of these simulations is summarized in Table 1.

RESULTS

The results of each simulation will be considered sequentially, followed by several key comparisons between selected simulations. Panel A of Table 2 and Row 1 of Table 3 show the numerical values associated with the independent variables. The justification for selecting these initial values is explained in the "Configuration and Conditions" column of Table 1.

Historical Simulation Versus Historical Data (Validation)

Panel A of Table 2 presents the means of the specified independent variables across six AQC classes. (This was discussed in the Validate Model section.) These database obtained values were used to initialize the independent variables for the Historical Simulation. Panel B of Table 2 shows the output (dependent variables) of the Historical Simulation and output from three of the six AQC classes initially referenced.

The data in Panel B of Table 2 show that students in the three classes being considered received almost twice the amount of flight simulator time as the three hours recommended by the POI. Because simulator time increased, the number of hours accumulated in the aircraft was lower than recommended, while the sum of these two remained near normal. Subject matter experts (SMEs) explained this discrepancy stating that

Table 1

Summary of Simulations

NAME	CONFIGURATION & CONDITIONS	PURPOSE
POI Baseline Simulation	Existing Configuration Student/IP ratio, # of students per class, # IPs and aircraft available set to POI guidelines Chance of inclement weather set at its historical value Unsatisfactory evaluation flight outcomes set to SME interview values	Analyze AQC under POI specified conditions Contrast with validation simulation
Historical Simulation	Existing Configuration Student/IP ratio, # of students per class, # IPs and aircraft available, and chance of inclement weather set to historical values Unsatisfactory evaluation flight outcomes set to SME interview values	Validation of model Analyze AQC as it is currently implemented
Experimental Simulation	Alternate Configuration Student/IP ratio, # of students per class, # IPs and aircraft available, and chance of inclement weather set to historical values Unsatisfactory evaluation flight outcomes set to SME interview values	Observe the effect this untested course design would have on makeup time

Table 2

Historical Simulation Versus Historical Data (Validation)

PANEL A
Independent Variables

Student/IP Ratio:	2.2:1
Students Per Class:	20
IPs Available (AM or PM):	14
Aircraft Available:	25
Inclement Weather Probability:	.05
Unsatisfactory Evaluation Flight Outcomes and Probabilities:	
Extra Hours Without Setback:	.15
Setback Without Extra Hours:	.02
Extra Hours and Setback:	.105
Dropped From AQC:	.015

PANEL B
Dependent Variables

	Historical Simulation	Database Information		
		Class 86-x7	Class 86-x8	Class 86-x9
Scheduled Flight Hours:	27.8	29.4	26.1	25.8
Makeup Flight Hours:	5.2	1.2	2.8	2.3
Scheduled Simulator Hours:	2.7	5.4	5.9	5.5
Makeup Simulator Hours:	.3	0	0	0
TOTAL HOURS:	36.0	36.0	34.8	33.6
Number of Students Given Unsatisfactory Evaluation Flight Outcomes:				
Extra Hours:	2	1	0	2
Setback:	2	0	1	0
Dropped From AQC:	0	0	0	0

Table 3

Independent and Dependent Variables By Simulation

---INDEPENDENT VARIABLES---

		COLUMN A POI Baseline Simulation	COLUMN B Historical Simulation	COLUMN C Experimental Simulation
ROW 1	Student/IP Ratio:	2:1	2.2:1	2.2:1
	Students Per Class:	26	20	20
	IPs Available (AM or PM):	20	14	14
	Aircraft Available:	18	25	25
	Inclement Weather Probability:	.05	.05	.05
	Unsatisfactory Evaluation Flight Outcomes and Probabilities:			
	Extra Hours Without Setback:	.15	.15	.15
	Setback Without Extra Hours:	.02	.02	.02
	Extra Hours and Setback:	.105	.105	.105
	Dropped From AQC:	.015	.015	.015

---DEPENDENT VARIABLES---

ROW 2	Scheduled Flight Hours:	30.2	27.8	27.8
	Makeup Flight Hours:	2.8	5.2	3.7
	Scheduled Simulator Hours:	3.0	2.7	4.1
	Makeup Simulator Hours:	0	.3	.4
	TOTAL HOURS:	36.0	36.0	36.0
	Flt Periods Cancelled(Weather)	2	1	1
	Flt Periods Cancelled(Resources)	3	4	6
	Flt Periods Degraded(Resources)	5	4	4
	Unsatisfactory Evaluation Flight Outcomes:			
	Extra Hours Without Setback:	Stu=0 Total Hrs=0	Stu=2 Total Hrs=8.0	Stu=2 Total Hrs=2.2
	Setback Without Extra Hours:	Stu=0	Stu=1	Stu=1
	Extra Hours and Setback:	Stu=2 Total Hrs=2.5	Stu=1 Total Hrs=5.2	Stu=1 Total Hrs=4.6
	Dropped From AQC:	Stu=0	Stu=0	Stu=0

students can graduate from the AQC with less than the POI recommended aircraft flight time under special conditions. Considering this explanation together with the actual class data, the Historical Simulation compares favorably in all categories. The number of students affected by an unsatisfactory evaluation flight also compares favorably between the simulation and reality, and therefore supports the model's validity.

POI Baseline Simulation

Row 2 of Columns A in Table 3 shows that the existing configuration and its associated POI defined independent variables, coupled with the historically low probability of inclement weather (.05), allowed students under these hypothetical conditions to receive more regularly scheduled flight time than students in any of the other simulations. The 2:1 student/IP ratio permitted each student to receive their total amount of time in the flight simulator. Repeated simulations using these independent variable settings produced dependent variable values similar to this run, thereby supporting the reliability of the model.

Historical Simulation Versus POI Baseline Simulation

Row 2 of Column B in Table 3 is an expansion of the results of the Historical Simulation (previously summarized in Panel B of Table 2). Due to the 2.2:1 student/IP ratio which is currently implemented, students did not receive their total allotted time in the flight simulator during regularly scheduled simulator periods. Students in the POI Baseline Simulation received the total allotted time due to the 2:1 student/IP ratio.

The primary difference between the results of the POI specified simulation, and those based upon historical data was that the POI runs accumulated more aircraft flight time and simulator flight time during scheduled training periods, and that makeup time was minimized. This outcome is primarily attributable to the 2:1 student/IP ratio in the POI simulation. These simulations show, therefore, that were the POI implemented as intended, a reduction of costly makeup time and, consequently, increased efficiency would be realized in this AQC.

Experimental Simulation

Data from the Experimental Simulation, in which the alternate configuration used historical data and a .05 chance of inclement weather, is shown in Row 2 of Column C in Table 3. This simulation resulted in less makeup time than the Historical Simulation but more than in the POI run. The Experimental Simulation had six flight periods cancelled due to lack of resources whereas the Historical Simulation had four cancellations. These findings suggest that the alternate configuration is not a viable alternative to the existing configuration (using either POI or historical independent variable values) because resources are not more efficiently allocated.

DISCUSSION

Model Reliability

The use of a stochastic model demands some check on the model's capacity to generate similar results when repeatedly run under identical conditions. Two separate tests were performed. The first was the replication of the simulation under POI specified conditions. This replication resulted in no substantial differences in course outcomes. The second was a direct examination of the values produced by the random number generator. A plot of these values (not shown) revealed no substantial deviation from normality. Both these outcomes support the model's reliability.

Model Validity

It has been suggested (Garzia and Garzia, 1986) that one method to validate a model is to compare the outcomes generated by the model when it has been initialized to simulate known conditions, with the outcomes obtained in the real system under the same conditions. In the AQC model, comparisons between the outcomes of the actual course and the simulation outcomes showed no substantial differences. The validity of the model is therefore supported.

Summary of Findings

The two course configurations (i.e., the existing and the alternate configurations) examined in this study were paired with three sets of conditions defined by historically based, POI based, and experimental independent variables. The model demonstrated the feasibility of using simulation techniques to test the effects of various course configurations and conditions. Beyond this demonstration, however, the simulations revealed an aspect of AQC operation which is of particular interest. Specifically, the chronic shortage of IPs prevents implementation of the POI as intended, and has the effect of reducing scheduled flight time and increasing makeup flight time. The overall result of this is to increase substantially the operating costs of the AQC.

It was also found that the changes introduced in this study which were expected to improve AQC efficiency, when simulated, did not result in significant improvements. Specifically, when simulator time was increased by 1.5 hours and flight time was decreased by 1.5 hours, no gain in resource allocation efficiency was realized. The absence of a gain is also directly attributable to the shortage of IPs. This unexpected outcome reveals the utility of devising a method (such as simulation) to test the effects of course changes prior to their implementation.

REFERENCES

Flight Training Guide for an Aviator Qualification Course. Aviation Training Battalion, Fort Rucker, AL: U.S. Army Aviation Center.

Master Scheduling Sheet, United States Army Aviation Center Form 138. Aviation Training Battalion, Fort Rucker, AL: U.S. Army Aviation Center.

Program of Instruction for an Aviator Qualification Course. Aviation Training Brigade, Fort Rucker, AL: U.S. Army Aviation Center.

Adkins, G. & Pooch, U.W. (1977). Computer simulation: A tutorial. Computer, 10, 12-17.

Coffield, D. (1987). An automated flight trainer's management system (in preparation). Fort Rucker, AL: Universal Energy Systems, Inc.

Garzia, R.F. & Garzia, M.R. (1986). Discrete-event simulation. IEEE Spectrum, December 1986, 32-36.

Graybeal, W.J. & Pooch, U.W. Simulation: Principles and Methods. Cambridge, Mass.: Winthrop Publishers, 1980.

Contact: Carol A.W. Sprunger
Aviation Training Office
Universal Energy Systems, Inc.
P.O. Box 537
Daleville, Alabama 36301-2337
(205) 598-4173 or 598-2571

A COMPUTER AUGMENTED CURRICULUM

B. Wayne Walters

University of Southern Mississippi

ABSTRACT

This paper will describe a mechanism for allowing all students in a university community to become familiar with the operation of personal computers and to provide for an understanding of both general and specific computer applications as they relate to the students' academic studies. The program's key component involve providing support for the faculty to learn to use personal computers.

BACKGROUND AND NEED:

The technical complexities of today's society are placing more demands upon professionals in completing their day-to-day activities. To stay competitive, they are being forced to use these new technologies. The development of the personal computer has added yet another layer of knowledge and skill for most of our educated population to master. The computer skill level required of the average educated person is increasing at a rapid rate. The use of personal computer based word processing, data base, and spreadsheet software is becoming commonplace.

A need for computer user skills, or computer technology, exists for all disciplines within a university just as it does outside of the university. The development of a campus-wide computer usage skills program for a university can provide many of the computer skills that businesses are requiring as well as serving the involved disciplines. The computer must be viewed as yet another tool for the student to master. The development of such a program must have the support and involvement of the faculty. At the present time, many academicians have not become involved in the current computer phenomenon. This may have been by choice or by chance. In most cases this has been because a framework has not existed to provide on-going support for them in developing these new skills.

AN APPROACH FOR INVOLVEMENT IN COMPUTING:

Recognizing this need, the University of Southern Mississippi has developed a campus-wide program to assist all disciplines in becoming more aware of computer technology. The program, which has become known as the Computer Augmented Curriculum (CAC) program, began in January of 1985. Today the program spans every discipline on campus. The program is organized with a director and 10 coordinators representing the 10 schools and colleges of the university. The coordinators are faculty members that are released from some of their academic responsibilities. They are responsible for helping their colleagues within their college or school with using personal computers in their teaching responsibilities. In return, the coordinators are provided with a personal computer. The director of the program is from the Computer Science department and has a two year appointment to the program. The director is available as a resource for the coordinators.

In May of 1985 an open bid process was initiated. There were 13 vendors that responded to the RFP. The Tandy 1000 was selected based upon their aggressive pricing, compatibility with the industry standard IBM PC, the stability of their corporation, and the availability of service support. An order was placed for 132 of the Tandy personal computers and they were delivered immediately. Microcomputer workcenters were created in 11 locations with 94 of the computers. The remainder of the systems were distributed to the coordinators for their use and for use as short-term loaners systems for the faculty. The CAC microcomputer workcenters are operated on an open basis for all students on campus. Additionally there are other microcomputer workcenters in other locations not associated with the CAC program.

The emphasis of the CAC program has been to involve the faculty and to let the faculty involve the students. The university recognized that this bootstrapping process would require making computers available for faculty use but did not feel that it could financially justify instituting a program to provide all faculty members with a personal computer. To obtain access to personal computers, faculty members may obtain a short-term loaner system from their coordinator or purchase their own computer from the university bookstore. The loaner computers are not for research but are intended for direct support of the curricula. The faculty need personal attention to address their specific needs. These needs can best be addressed by knowledgeable people within their academic area. Coordinators, with support from the director, have conducted training sessions with the faculty on various aspects of personal computer use and are available for consultation with the faculty for developing and using courseware material.

The university bookstore has played an active role in the program by creating a computer outlet for selling personal computers, software, and accessories for faculty, students, and staff. The store carries IBM, Zenith, and Apple computer systems. Faculty can also obtain Tandy products at the local Radio Shack store at 20% off list price. The prices offered by the bookstore's computer outlet are often competitive with discount mail-order outlets. In addition, the university provides financing arrangements for faculty and students to aid them with the purchase of their own personal computer systems.

SOFTWARE PROVISIONS:

Software is another key ingredient of a program such as the CAC. The emphasis of the program has been on the industry standard IBM personal computer or clones with MS-DOS compatibility. In many respects the demand for software products is greater than the demand for hardware. Commercial software has been found to be too expensive for widespread use by the faculty and it poses a copyright liability concern when used with a large student population. When possible, the university has depended upon public domain/shareware software. Experience has shown that programs such as PC-Write are more than adequate in meeting the word processing needs of a university environment. PC-Write has become the defacto standard for word processing on campus. Software packages such as PROCOMM or QMODEM are examples of communication packages that are being used. In addition, locally developed programs, such as PC-Grade, are widely used for grade book management. Where these products have not been adequate, low-cost alternates have been used. For example, VP-Planner is used as a low-cost alternative for Lotus 1-2-3.

Software copyright protection is an ethical concern for all professionals. It is important that universities address this issue from an ethical as well as a legal posture. University faculty and staff cannot be expected to police the use of equipment and software on campus but can help to set the tone for acceptable behavior. In an attempt to address the legal issue of software copyrights, the university has adopted a policy on software copyright protection modeled after the ADAPSO statement.

CURRENT STATUS:

The CAC program has been through two stages and is currently entering its third stage. The first two stages involved the creation of the personal computer clusters and the two year operational phase under the CAC director's management. The third phase begins with the elimination of the director's position. The third phase will provide new challenges for the program since a director will not be available to provide a central focus and control. Instead, the CAC coordinators are to be organized into a governing board to oversee the operation of the program. The responsibilities of the board are to enforce the standards and policies established for the labs, develop and submit requests for new resources to support the program, and to develop budget requirements for operating the CAC labs. The board is to report to the Vice President for Academic Affairs.

The challenge for the future will be maintaining the standards which have been developed and to develop new standards when appropriate. The program is currently run on MS-DOS, IBM PC compatible computers. The transition to a new computing environment to meet changing technologies will be critical to maintaining a viable, on-going program.

Likewise, maintaining some degree of software standardization will be challenging for the new governing body.

SUMMARY

The goal of the Computer Augmented Curriculum at the University of Southern Mississippi is to provide a mechanism for allowing all students to become familiar with the operation of personal computers and to provide for an understanding of both general and specific computer applications as they relate to the students' academic studies. The University of Southern Mississippi has provided the necessary computing equipment and created microcomputer workcenters. It also has provided a framework for implementing the program. The success of the program now resides in the hands of the faculty.

REFERENCES

1. Association of Data Processing Service Organizations (ADAPSO), *Policy statement*,"Thou Shall Not Dupe", 1985

2. *Directory of Public Domain Software for the IBM Personal Computer*, PC Software Interest Group, Santa Clara, Ca., 1984

3. Gader, Bertram and Nodar, Manuel, *Free Software for the IBM PC*, Warner Books Inc., New York, NY. 1984

4. Wallace, Bob, *PC-Write*, Quicksoft, Seattle, Wa.

Improving the Measurement of Musical Ability Through Adaptive Testing

Walter P. Vispoel

University of Iowa, Iowa City, Iowa

Abstract

This study compared the reliability, validity, and efficiency of a newly developed computer administered adaptive test of musical ability and conventional fixed-item tests of similar content. Adaptive test validities and reliabilities across all ability levels exceeded those of the conventional tests. Most dramatically, the adaptive test reached reliabilities comparable to one of the conventional tests using an average of 72% fewer items.

Introduction

With the advent of item response theory and recent advances in computer technology, adaptive testing is now feasible in most subject matter areas (Lord, 1980). Adaptive testing software like MICROCAT from The Assessment Systems Corporation and packages developed by the WICAT Systems Company are commercially available. Although these products are important contributions to modern testing technology, they require that test items be in written form. This paper describes data from an adaptive test that uses aural stimuli (i.e.; musical pitches) to assess musical ability. Data derived from this test indicates that adaptive tests of musical ability can overcome problems of poor reliability, validity, and efficiency that are common in conventional fixed-item tests. (See Shuter-Dyson & Gabriel, 1981, Whellams, 1971, and Vispoel, 1986 for discussions of the shortcomings of commercially available musical ability test batteries.)

Background

All commercially available tests of musical ability share one characteristic: They use fixed set of items that are administered to all examinees regardless of their ability level. Since examinees generally differ in the ability being measured by a given test, some items will be too easy and others too difficult to be informative about a given person's abilities. Nonetheless, examinees may devote considerable time and effort to answering these items. Adaptive testing procedures are aimed at improving test reliability and efficiency by successively presenting items of appropriate difficulty to each examinee according to performance on previous items. Appropriate items are selected from a necessarily large pool that contains sufficient items with which to estimate each ability level with high precision. A major benefit of adaptive testing is that it reduces, substantially, the number of test items required to reach a high level of measurement precision for most examinees. Moreover, ability levels are measured with greater reliability because each examinee receives a greater proportion of appropriate items. Increasing test reliability in turn is likely to enhance test validity.

Test Development

The first step in creating the present adaptive test was to develop the item pool. Test items were similar in format to those used in the Seashore (see Seashore, et al, 1960), Wing (1968), and Bentley (1966) Tonal Memory Tests. A typical test item requires an examinee to compare two versions of a short melody. On the second playing, either the same melody is repeated, or one note is altered. Examinees indicate the number of the note that has changed or indicate that the two playings of the melody are the same.

Three hundred and fourteen pilot items were developed. A subset of 60 or 104 items of these items was administered to 468 examinees. Item response theory difficulty and discrimination indices for the 314 items were estimated using the computer program LOGIST (Wood, Wingersky, & Lord, 1976). This program automatically puts item difficulty and discrimination indices on common metrics if all examinees respond to a common subset of items. This was accomplished in the present case by administered 12 common items to all examinees. One hundred and eighty items from the original 314 were chosen for the final adaptive test item pool. High fidelity measurement across all ability levels was ensured by selecting highly discriminating items at all levels of item difficulty. (See Vispoel, 1986 for further details.)

The adaptive test was evaluated by comparing it to conventional music tests in terms of reliability, validity, and efficiency.

Test Reliability

The reliabilities of the adaptive test and Seashore Tonal Memory Test items are compared across ability levels in Table 1. The ability scale in Table 1 is labeled Theta (θ), the conventional item response theory ability scale symbol. For purposes of this discussion, θ scores can be interpreted as Z scores since they have a mean of θ and a standard deviation of 1 (see Lord, 1980 for a more detailed explanation of θ scores and item response theory). Test reliability was assessed using Urry's (1977) "adaptive test" reliability coefficient.

As can be seen from Table 1, adaptive test items provide high measurement precision across all ability levels. Over the θ range of -2 to +2, which represents 95% of the ability distribution, reliabilities were .95 or higher. Even at the ability extremes of -3.5 and +3.5, reliabilities were .82 and .85, respectively. The Seashore test, which measures ability reasonably well within the θ range of -3.0 to +0.1 (i.e., reliability greater than or equal to .82), becomes increasingly unreliable as ability moves from medium to high levels. At $\theta = 1.2$ or higher, for example, the Seashore test has a reliability of 0.0. These results clearly illustrate the remarkable improvements in reliability that are possible through adaptive testing, as well as the difficulty in measuring all ability levels precisely using conventional fixed-item tests.

Test Validity

Validity coefficients for the Seashore Tonal Memory, Drake (1957) Musical Memory, and scores based on subsets of adaptive test items. Validities were assessed by correlating test scores with 1) examinees' perceptions of their general musical ability using a five point scale with 1 = low ability and 5 = high ability, 2) the number of years an examinee has played a musical instrument, and 3) whether an examinee ever played music professionally (1 = yes, 0 = no). As evident from Table 2, validities for adaptive test scores were higher than those for the Seashore and Drake Tests on each of the three validity measures.

It should be noted, however, that the validities of scores derived from adaptive test items are based on the items from the 180 item pool that each examinee from the original sample responded to. These examinees did not take the actual adaptive test. Ability scores derived from the adaptive test pool are based on fixed sets of 37 to 64 items. Since the Seashore and Drake tests contained a comparable number ot items, that is 30 and 53 respectively, rough comparisons of validity coefficients can be made. However, validities based on 30 to 50 items administered using the actual adaptive test are expected to be even higher. This is because items are matched to each examinee's ability level, thereby providing a more reliable and potentially more valid ability score.

Test Efficiency

In addition to enhancing test reliability and validity, adaptive testing can increase test efficiency dramatically. Table 3 indicates the number of items that were necessary to reach reliabilities of .80, .85, and .90 for 20 examinees who took the actual adaptive test. Note, in particular, the impressive overall efficiency of the adaptive test for the examinees whose abilities range from moderately low to moderately high (i.e., θ's from -1.36 to +.93). As shown in Table 4, in this range, an average of approximately 5, 6, and 8 adaptive test items was required for reliabilities of .80, .85, and .90, respectively.

The overall reliability of the 30-item Seashore test was .85 and comparable reliabilities on the adaptive test were reached by the sample of 20 examinees after an average of 8.55 items. This represents a 72% reduction in test length. Moreover, the efficiency of the present adaptive test at low to moderately high ability levels matches or surpasses the efficiencies of any of the adaptive tests described in the research literature (Urry, 1977; McBride, 1979; Weiss, 1982).

Conclusion

The present results indicate that an adaptive test of musical ability is more reliable and valid than a conventional fixed item test of similar content. Most convincingly, these improvements in test reliability and validity can be accomplished with substantially fewer test items. These findings strongly suggest that the development of additional adaptive tests could provide significant improvements in the measurement of musical ability.

References

Bentley, A. (1966). *Musical ability in children and its measurement*. London: Harrap.

Drake, R. K. (1957). *Manual for the Drake Musical Aptitude Tests*. Chicago: Science Research Associates.

Lord, F. M. (1980). *Applications of item response theory to practical testing problems*. Hillsdale, NJ: Erlbaum.

McBride, J.R. (1979). *Adaptive verbal ability testing in a military setting*. In D.J. Weiss (Ed.), *Proceedings of the 1979 Computerized Adaptive Testing Converence*. Minneapolis: University of Minnesota, Department of Psychology, Psychometric Methods Program, Computerized Adaptive Testing Laboratory.

Seashore, C. E., Lewis, P. & Saetveit, J. C. (1960). *Manual of instructions and interpretations for the Seashore Measures of Musical Talents (2nd revision)*. New York: The Psychological Corporation.

Urry, V. W. (1977). Tailored testing: A successful application of latent trait theory. *Journal of Educational Measurement*, 14(2), 181-196.

Vispoel, W. P. (1986). *An adaptive test of musical memory*: *An application of item response theory to the assessment of musical ability*. Unpublished Doctoral Dissertation, University of Illinois at Urbana-Champaign.

Weiss, D. J. (1982). Improving measurement quality and efficiency with adaptive testing. *Applied Psychological Measurement*. *6*, 473-492.

Whellams, F. (1971). *The aural musical abilities of junior school children*: *A factorial investigation*. Unpublished Doctoral Dissertation, University of London.

Wing, H. D. (1968). Tests of musical ability and appreciation (2nd edition). *British Journal of Psychology Monograph Supplement*.

Wood, R. L., Wingersky, M. S., & Lord, F. M. (1976). *LOGIST: A computer program for estimating examinee ability and item characteristic curve parameters* (Research Memorandum 76-6). Princeton, NJ: Educational Testing Service.

Table 1: Comparisons of Adaptive Test and Seashore Items on Reliability Indices for Theta (θ) from -3.5 to 3.5.

Theta (θ)	Reliability Adaptive Test	Seashore
-3.5	.82	.50
-3.4	.84	.59
-3.3	.86	.66
-3.2	.88	.72
-3.1	.90	.77
-3.0	.91	.82
-2.9	.92	.85
-2.8	.93	.88
-2.7	.94	.90
-2.6	.95	.92
-2.5	.95	.93
-2.4	.96	.94
-2.3	.97	.95
-2.2	.97	.96
-2.1	.97	.96
-2.0	.98	.97
-1.9	.98	.97
-1.8	.98	.97
-1.7	.98	.97
-1.6	.99	.97
-1.5	.99	.97
-1.4	.99	.97
-1.3	.99	.97
-1.2	.99	.96
-1.1	.99	.96
-1.0	.99	.96
- .9	.99	.95
- .8	.99	.95
- .7	.99	.94
- .6	.99	.94
- .5	.99	.93
- .4	.99	.92
- .3	.99	.90
- .2	.99	.89
- .1	.99	.87
0.0	.99	.84
.1	.99	.82
.2	.99	.78
.3	.98	.74
.4	.98	.70
.5	.98	.64
.6	.98	.58
.7	.98	.51
.8	.98	.42
.9	.98	.33
1.0	.98	.22
1.1	.97	.10
1.2	.97	0.0
1.3	.97	0.0
1.4	.97	0.0
1.5	.97	0.0
1.6	.96	0.0
1.7	.96	0.0
1.8	.96	0.0
1.9	.95	0.0
2.0	.95	0.0
2.1	.95	0.0
2.2	.94	0.0
2.3	.94	0.0
2.4	.93	0.0
2.5	.93	0.0
2.6	.92	0.0
2.7	.92	0.0
2.8	.91	0.0
2.9	.91	0.0
3.0	.90	0.0
3.1	.89	0.0
3.2	.88	0.0
3.3	.87	0.0
3.4	.86	0.0
3.5	.85	0.0

Table 2: Correlations of Adaptive Test, Seashore Tonal Memory and Drake Musical Memory Scores with Selected Validation Measures.

	Correlations		
Validation Measure	Adaptive Test	Seashore	Drake
Self Perception of General Musical Ability	.46	.36	.31
Yrs. of Instrument Playing	.34	.24	.27
Professional Experience	.44	.21	.33

No. of Examinees=223

Table 3: Number of Adaptive Test Items Needed to Reach Selected Reliabilities for Examinees of Selected Ability Levels.

Examinee Number	θ	Reliability ($r_{\theta\hat{\theta}}^2$) .80	 .85	 .90
		Number of Items Administered		
1	-2.51	16	17	25
2	-2.41	16	19	20
3	-1.36	5	6	7
4	-1.22	5	6	7
5	- .73	4	5	6
6	- .62	4	6	9
7	- .57	5	5	7
8	- .51	6	7	11
9	- .32	4	6	8
10	- .24	6	7	11
11	- .23	4	5	6
12	- .20	4	5	7
13	- .03	4	5	8
14	- .03	4	5	8
15	.09	4	6	8
16	.40	4	5	8
17	.93	5	5	13
18	.93	8	9	14
19	1.47	13	18	25
20	1.89	12	18	31

Table 4: Average, Minimum, and Maximum Number of Items Needed to Reach Selected Reliabilities for Total and Restricted Samples.

	Restricted Sample n = 16 range = -1.36 to +.93 Number of Items			Total Sample n = 20 range = -2.51 to +1.89 Number of Items		
	Average	Minimum	Maximum	Average	Minimum	Maximum
$r_{\theta\hat{\theta}}^2 = .80$	4.75	4	8	6.05	4	16
.85	6.19	5	9	8.55	5	19
.90	8.19	6	14	11.60	6	31

Contact: Walter P. Vispoel
University of Iowa
Iowa City, IA

ORIGINS AND EARLY DEVELOPMENT OF PILOT

John A. Starkweather

University of California, San Francisco

Abstract

Historical examples describe early applications of PILOT and the COMPUTEST language that preceded PILOT. Using a model of human conversation resulted in a choice of central mechanisms meeting many diverse needs in a variety of settings. The system has been adapted to operate on many different computers and has developed new capacity as computers have become more capable and interactive. This review emphasizes early development.

The PILOT language developed as a result of experience with an interactive computer system called COMPUTEST (1), developed at the University of California in San Francisco in 1962. This system ran on an IBM 1620 computer which had 20,000 character of memory, a card reader and punch, and a console typewriter that was used for control by the operator. COMPUTEST was able to carry out interactive programs with one user at a time by means of its console typewriter. It was distributed by IBM as part of the 1620 General Program Library in 1965. In early versions the COMPUTEST system operated with program information (the program written in COMPUTEST) stored in a deck of punched cards, and read into the computer a few at a time when the program was executed. Later, extended versions of the system had greatly increased capabilities by using disk storage units that allowed random access to the program material. Similar disk-based versions were developed by Bill Turner at UCSF and Jim Grisell at the Lafayette Clinic in Detroit, Michigan. These versions used an IBM 1311 disk drive which was about the size of a washing machine and stored 5 megabytes. (Samples of program material will be shown developed by and for elementary and high school pupils as well as medical students).

The central mechanisms of COMPUTEST, and later PILOT, were chosen as those necessary for interaction between a computer and its user in a conversational manner. In the days of punched cards and batch systems, sometimes with no better that 24 hour turnaround, it was unusual to think of using a computer in this manner, but the IBM 1620 computer had a console typewriter that made it possible. The basic mechanisms chosen with this conversational model in mind have proven useful in many settings and with a variety of different computers.

When an IBM 360 model 50 computer with time-sharing capability became available at the UCSF campus in 1967, a new computer-assisted instruction system called PILOT (2) was developed. Dave Gomberg was the primary programmer for a version of PILOT written in PL/1. The language features of the PILOT system were patterned after those of COMPUTEST and stressed easy entry of instructional material by the program author. We took a fresh look at what functions were most useful after review of 234 COMPUTEST programs produced by elementary school children during a year's use of the 1620 computer. We felt that a teacher should not have to become a computer expert in order to develop a computer-assisted instructional program, and we were impressed that students often learned a great deal by developing instructional or testing sequences. The PILOT system began to be used for traditional frame-oriented instructional programs as well as providing practice with simulated clinical problems, self-evaluation testing, and simulated interviewing (3,4) with natural language input from students.

Not long after the original version of the PILOT language was written for the IBM computer, several versions of the language were programmed to run on other computers that had interactive capability. 940-PILOT was developed by David Ladd, Tom Baumbach, and Richard White in the Electrical Engineering Department at the University of California, Berkeley. In 1968 it was written in Assembly language for an experimental computer system called Project Genie. This computer was a prototype of the SDS-940 time-share system that formed the basis of TYMNET.

Sylvan Rubin and Dean Brown of the Stanford Research Institute used the SDS-940 version of PILOT in an education research program and found that both teachers and students learned it easily and could write interesting programs immediately. In 1971 Rubin wrote a smaller version as an interpreter in BASIC and called it PYLON. Other versions of PYLON or PILOT were produced at Stanford University, the Northern California Regional Computer Network of about 18 colleges, and by Pete Rowe at the Lawrence Hall of Science of the University of California, Berkeley. Also in 1971 Dick Martz and Arthur Luehrmann developed a PILOT interpreter at Dartmouth College that was written in their time-sharing system BASIC, using GE-635 equipment. (Examples of programs written in early PILOT for different computers and in different subject matter areas will be shown).

Partly because of the requirements of different computer systems, and partly because of individual preferences, each version of the language was somewhat different from the others. In early 1973, representative of six of the major versions of the PILOT language met together and agreed on a set of core language specifications to be common among all the systems. This language represented the experience of many workers in computer-assisted instruction at that time.

Recognizing that differences would undoubtedly arise, there was also agreement on a standard means of describing functions outside the core language. Because many of those involved had applications in elementary and secondary school settings, the resulting standard, called PILOT 73, was intended to be a specialized language that would be easy for program authors to learn. Such authors were imagined to be teachers without prior knowledge abut computers.

The PILOT 73 standards allowed only eight basic codes. These were single letters chosen to be mnemonic, such as T for type, A for accept an answer, and M for match. Extensions to the core language were to be words rather than codes requiring further memorization. Rubin summarized the PILOT 73 standards in a brief paper that further prompted implementations on large and small computers (5). PILOT was seen as a reasonable means for translation of computer instructional programs from one language to another. For example, Fraser, at the University of Washington in Seattle, used a translator from Coursewriter to PILOT in order to run existing course material as part of an interactive system using the MUMPS language. Lawrence Turner, in the Physics Department of Pacific Union College, implemented PILOT 73 in BASIC for Hewlett-Packard equipment and this was published by Hewlett-Packard as part of a computer curriculum project (6).

Charles Tidball of George Washington University in Washington, D.C. developed PILOT for large IBM equipment running the time-sharing system TSO and a PILOT program called MEDLEARN that has now been used for many years to train medical librarians in the use of the National Library of Medicine's MEDLINE system. He later developed a version for the PDP-8 minicomputer.

At UCSF we were interested in avoiding the need for users to dial-in to a remote computer. Although this was prior to the availability of microcomputers, the Datapoint 1200 desktop computer provided a means to investigate self-contained operation and avoid the problems of communication with early time-sharing systems (7). This machine had a built-in minicomputer, two magnetic cassette tape drives, a keyboard and cathode ray tube for character display of 12 lines of 80 characters. The memory capacity of the first of these machines was 8000 bytes. About three quarters was used for the PILOT which left room for about 2000 characters of program material at a time. This was enough to hold a number of instructional frames in active memory and additional material was read from tape when it was needed. The tape could contain about 100,000 characters of program material, and was sufficient for the development of modest elements of interactive learning to be used as an adjunct to classroom instruction.

A major dialect of PILOT that departed from some of the PILOT 73 guidelines was developed by George Gerhold and Larry Kheriaty at Western Washington State College in Bellingham, Washington. It was called COMMON PILOT, and had extensions and additional functions to handle more complex programming needed in college level instruction, particularly in mathematics and science. This version of PILOT formed the basis of PILOT for the Apple microcomputer (8), for Radio Shack microcomputers (9), and PC/PILOT for the IBM Personal Computer (10).

PILOT 73 for the Datapoint computer became the basis of versions for an early microcomputer, the Processor Technology Sol. Before this company went out of business, they published a cassette version of PILOT (11) and a version for their Helios disk system was in preparation. These were later distributed by PROTEUS, The Processor Technology Users Group (12).

In 1976 Charles Goldstein of the Lister Hill National Center for Biomedical Communications, the research and development branch of the National Library of Medicine, initiated a computer conference on TYMNET to develop ideas about improvements to PILOT. Those who were participants in the conference will remember the difficulty in keeping abreast of commentary from all the writers on various topics and they will perhaps recall the differences between those arguing for simplicity versus the need for extensions. One outcome of the conference was a set of specifications that resulted in a contract let by the National Library of Medicine to Boeing Computer Services for LHC-PILOT (13). One of the authors, William Ford, later organized a new company, Online Systems, to produce a commercial version of PILOT called PILOT-PLUS.

Atari, Inc. developed PILOT for its microcomputer in a version that is easy for elementary students to use (14). It contains graphic extensions providing the kind of "turtle" graphics found in LOGO. Although PILOT itself does not use line numbers, the Atari version requires them for program entry and editing, using the same mechanisms as provided for BASIC.

References:

1. Starkweather, J.A. Computest: a computer language for individualized testing, instruction, and interviewing. Psychol. Rep., 1965, 17, 227.

2. Starkweather, J.A. A common language for a variety of conversational programming needs. Readings in Computer-assisted Instruction, H.A. Wilson and R.C. Atkinson (Eds.), New York: Academic Press, 1969, p. 269.

3. Starkweather, J.A., Kamp, M., & Monto, A. Psychiatric interview simulation by computer. Meth. Inform. Med., 1967, 6, 15.

4. Kamp, M. Evaluating the operation of interactive free-response computer program. J. Biomed. Systems, 1971, 2, 33.

5. Rubin, S. A simple instructional language. Computer Decisions, November 1973, 17.

6. Turner, L.E. PILOT Reference Manual, Hewlett-Packard Company, Cupertino, CA 94017, 1973.

7. Kamp, M. & Starkweather, J.A. A return to a dedicated machine for computer-assisted instruction. Comp. Biol. Med., 1973, 3, 293.

8. Apple Pilot and Apple Super Pilot Apple Computer, Inc., 10260 Bandley Drive, Cupertino, CA 94017, 1980.

9. TRS-80 Pilot Radio Shack Education Division, 1600 One Tandy Center, Fort Worth, TX 76102.

10. PC/PILOT Language Reference Manual Washington Computer Services, 3028 Silvern Lane, Bellingham, WA 98226.

11. Cassette PILOT Users's Manual, Pleasanton, CA: Processor Technology Corp., 1978.

12. Disk-PILOT Reference Manual PROTEUS, (Processor Technology Users' Group), 1690 Woodside Road, Suite 219, Redwood City, CA 94016, 1979.

13. Chapman, G.I. and Ford, W.H. LHC-PILOT User Guide and Systems Guide, Lister Hill National Center for Biomedical Communications, National Library of Medicine, 1978.

14. Atari Pilot Atari, Inc. 1272 Borregas Avenue, Sunnyvale, CA 94086, 1981.

GENERATIVE CAI FOR LATIN: ASSESSING A TEN-YEAR PROJECT

Gerald R. Culley

University of Delaware

ABSTRACT

A ten-year project of generative CAI in Latin has produced about one hundred hours of instruction. Despite its gains in efficiency, sophistication of judging, and precision in review, the generative approach is essentially concerned with morphology, not meaning. To utilize its full potential as a language teaching tool, one must combine the generative approach with semantic knowledge.

Just over ten years ago, I began experimenting with generative CAI in Latin. This is rather a long time for one project, and in the volatile world of computers, it seems an eon. Time, then, to look back over the tangled mass of printouts and try to gain some perspective on the experience. Did generative CAI contribute to Latin instruction? And apart from what it may have taught my students, did I learn anything from it? That's what this paper is about.

Before attempting that assessment, though, I should explain just how I am using the term "generative CAI." In foreign language instruction, it means teaching the computer how to inflect the variable parts of speech and "know" the grammatical identity of each form. This approach is especially well suited to the highly inflected languages (such as Latin, Finnish, German, and Classical Greek). I would like to claim that I invented it, but pioneers like Hubert Weller (1976), Henry Decker (1976) and Robert Phillips (1981) were doing it for French and Spanish before I came along, and a Latin verb conjugation/verb analysis program was running in Paris in 1975 (Muller, 1975). And before all of these, Hugh Olmsted (1975) had published his work with generative CAI for Russian. It won't even do for me to claim that the technique is terribly difficult. About five years ago a popular computing magazine published (Lippman, 1982) the complete listing of a program to conjugate any regular Latin verb. The program was written by a high school student, David Lippman.

No, my part in all of this was just that I had access to exceptional institutional support at the University of Delaware. Originally that support organization consisted of a two-person staff called the PLATO Project; now there are some seventy full-time professionals and troops of part-time programmers making up what is now called the Office of Instructional Technology. Under any name, this organization always seemed to have resources available for me. So I had the luxury of doing what others could not--applying generative CAI to Latin instruction in a series of fairly elaborate programs, testing them, revising them, and eventually implementing them on both mainframe and micro systems.

Let me sketch that very briefly. With access to the PLATO system and as much technical advice as I wanted, I began developing and programming instructional lessons which used generative techniques. First came Verb Factory and Cursus Honorum, two verb lessons. The former is diagnostic quizzing with multiple levels of diagnostic help, and the latter a game format for practicing the production and recognition of verb forms. Next was Mare

Nostrum, a practice format for noun/adjective phrases. The fourth lesson was Translat, a program for Latin sentence translation which gives the student access to an array of lexical, grammatical and syntactic information as an aid to translation. Then came Artifex Verborum, in which the student does parsing of Latin words in sentence context, with wrong-answer feedback produced by the generative routines. After only six years and several thousand hours of work, I had five lessons on line, tested and revised. Then that same organization, called by that time the Office of Computer-Based Instruction, funded the conversion of these five lessons to Applesoft BASIC for the Apple II series of microcomputers (Culley, 1984, 1984-86). A bit over two years later, when that project was winding down to a successful conclusion, I brought out a sixth lesson, Lector. It allows students to search through Latin text for some specific feature of the language (like direct objects), and it uses the generative routines in providing various kinds of corrective feedback when the student goes astray. The Office of Instructional Technology also supported the conversion of this lesson from PLATO to the Apple II series. That program made its appearance this summer (Culley, 1987).

Last spring, OIT approved my proposal to incorporate a major enhancement into the six *Latin Skills* programs and convert them to IBM PC format. The two most recent programs, Artifex Verborum and Lector, will be enhanced so as to accommodate a much larger database of Latin. In their present form, they contain about four hundred relatively short Latin sentences, and the maximum sentence length is only fifteen words. The new format will accept unaltered Latin, which may contain sentences of well over one hundred words. The plan calls for these programs to access a database containing four books of Vergil's *Aeneid* and four of the speeches of Cicero. The timeline for the project is another two years.

That's the story to date. About a hundred hours of instruction have resulted from these ten years of development. Thousands of students have been exposed to the programs. This is a good time, then, to comment on what generative CAI has and has not been able to do.

First, let's see the advantages of the generative approach.

1. A vastly expanded number of exercises can be generated without corresponding increases in the memory requirement. Since the computer inflects the forms on demand, a small database of stems has maximum impact. A program on noun-adjective phrases with 100 noun bases and 80 of adjectives yields 8,000 possible pairings, each of which may appear in one of ten forms (five cases, singular or plural). There is therefore a potential for 80,000 drill items.
2. It becomes possible to offer much more sophisticated judging of student answers and feedback on partially correct ones. A program that can conjugate a verb can be made to separate a student's typed verb into stem, tense/mood sign, and personal ending and comment on it in those terms. The following examples illustrate some of the power of the technique.

a. Give the Latin for "We were walking."

Student Types	Computer Responds
ambalat	The stem is wrong.
ambulabit	Check the stem vowel/tense-mood sign.
ambulabat	The personal ending is wrong.

b. Give the Latin for "She will hear."

Student Types	Computer Responds
audibit	Fourth conjugation verbs don't use **-bi-** in the future.
audiemus	The form you typed is FIRST person PLURAL future active indicative.

c. Give the Latin for "He has been helped."

Student Types	Computer Responds
adiuta est	The form you typed is third person singular perfect passive indicative, FEMININE gender.

The pedagogical potential of such morphemic analysis is obvious.

3. A very nice kind of corrective feedback can be done where students must identify a form. Suppose a verb like **amabam**, I loved, is to be parsed. The student marks items on a grammatical chart to identify the form. Let's suppose he identifies it incorrectly: third person singular, future active indicative (rather than imperfect). The program then takes

this input and uses it to inflect the verb stem again. The result is **amabit**, he will love; this is displayed beside the target form. Each time the student inputs a set of parameters, he sees the result this produces; a kind of mini-lab for Latin morphology is created.

4. Another advantage is that "invisible" review is possible. By that I mean the capability to review missed forms by presenting, three or four items later, not the item that was missed, but a similar one. For example, a student misses **muniebat** in a drill. Three items later, by giving him **dormiebat**, the computer reviews the missed concept (third singular imperfect active indicative of the fourth conjugation) rather than the item. Such review is often invisible to the user, and thus more effective, but it can only be done if the program "knows" the grammatical identity of the items presented.
5. A generative program can be both generic and specific at the same time. That is, its heart is the set of routines which inflect the variable parts of speech. Once those routines are in place, it is only necessary to feed in a set of stems and the program can produce inflected forms. The forms can either be displayed as items for the student to identify, or concealed to serve as the answers to questions the program generates. The program driver, then, is generic: it works for Latin, regardless of level of instruction, textbook in use, vocabulary known, or other variables. But the data, when processed by the driver, can be specific to the needs of the student. Feedback on a missed answer can be precisely focused, almost as if questions and answers had been hard-coded for one textbook, one set of vocabulary, and one instructional level. To a certain extent, the generative approach lets one have it both ways.
6. The fact that the programs have a generic quality, that only one driver need be written for a broad range of uses, means a considerable saving in resources. It becomes possible to spend much more time on a single program when it is known that the program will produce many hours of instruction. For example, Artifex Verborum, as developed on PLATO, devoured about a thousand hours of my time; but it yields about twenty hours of instruction for a given student. This translates into about fifty hours of development for each student hour, a quite acceptable ratio; but the dividend is that one can do a lot of rethinking, reworking, and polishing in that much time. The result is seen in higher quality overall. No secrets here: more time invested means more achieved.
7. But there is more to this saving of resources. That same program, Artifex Verborum, along with its editor, took about a thousand hours to develop, test, revise and polish. As I said, about a fifty to-one ratio of development time to instructional time. But that is for the first database, for textbook number one, if you like. The second database, and the third, come much cheaper. In fact there are now five databases for that program, one for each of five different textbooks. Now we are talking about a hundred hours of instruction from the program--a ratio of only a little over ten development hours for each hour of instruction.
8. The generative approach yields a number of different applications. When I had completed Verb Factory, it was no major task to use its verb-conjugating routines as the core of a new lesson, Cursus Honorum. They look very different to the student, but the most difficult algorithms are shared. Similarly, when Artifex Verborum was completed and tested, it seemed a shame not to capitalize on its successful elements. And so Lector was born, a lesson that shares not only the inflecting routines of Artifex, but its multiple databases as well. Yet the pedagogical goals and overall appearance of the two lessons are quite different.
9. Lessons designed around the generative approach are very well suited for collecting and analyzing data on student errors. One example of this is found in Verb Factory, which was set up to collect data on the kinds of errors students made as each new form of the verb was introduced. Since the program could distinguish among errors which involved the stem, the tense/mood sign, or the personal ending of the verb, it was not difficult to group these errors in the same categories. There are thirteen diagnostic exercises in the program, a new one for each time some new tense, voice or mood of the verb is introduced. Over the first four of these, the percentage of student errors attributable to the verb stem gradually decreases from 19% to 7%, as

one might expect, since the same stem is used throughout these four levels, while the tense/mood sign and personal ending are the locus of new morphemes. At the fifth test, however, the percentage of errors attributable to the stem more than triples. It is at this point that a new stem, the perfect active stem, makes its appearance. After four quizzes with forms like **agunt**, they suddenly encounter **egerunt**. Similarly, when passive forms are introduced--at test seven--the percentage of errors chargeable to the personal ending more than doubles. This clearly results from the novelty of the passive personal endings, which students have not yet mastered. They know **laudant**, but not **laudantur**. Despite, then, the instructor's valiant efforts, one can see that a new stem or new personal endings cause students to become confused. A look at these percentages makes it clear just how important it is to introduce new forms cautiously, clearly and coherently.

One additional example of data from generative programs is a confusion matrix from the other verb program, Cursus Honorum. The confusion matrix attempts to answer the question, "If a student sees a verb form and identifies it incorrectly, what form did he think it was?" The x-axis, at the top of the graph, lists all possible correct verb forms, while the y-axis lists the student responses. A diagonal line from upper left to lower right tracks correct responses. If, then, a verb form was third person plural imperfect active indicative, but the student identified it as third person singular future active indicative, the former would be plotted on the x-axis and the latter on the y-axis. The node where these intersect represents "thinking a third plural imperfect active indicative was a third singular future active indicative". The larger the number at any one node, the more common that particular error was. Examination of the confusion matrix shows a number of diagonal lines made up of these frequent-error nodes. They are "ghosts"--reflections of the correct-answer diagonal. For example, one such "ghost" shows that student frequently mistook future active indicative forms (like **ducent**) for present active indicative forms (like **ducunt**). Data of this kind support the view that these student errors are conceptual: they reflect a wrong mental model of the Latin verb. One tense is mistaken for another with some frequency, regardless of the person and number. One could infer from this that teachers need to place more emphasis on the pattern that can be seen in the Latin verb, rather than simply asking students to memorize individual forms.

The project had a significant impact on our instructional program. The whole series of programs, which has the title Latin Skills, was intended to supplement traditional instruction. Students are assigned to use the computer about an hour a week as a part of their homework. But it soon became clear that well-motivated students who used the programs could study Latin with very little supervision. We were able, then, to design a separate, self-paced track for the first year of Latin which relies heavily on the computer materials. Since its introduction five years ago, the program has increased our enrollments in elementary Latin by 70%. None of us foresaw that effect, but we are not displeased.

That self-paced program was, in fact, successful enough that it seemed worthwhile to distribute it too, and the University funded the task of packaging the material. It consists of a syllabus, a test-generating diskette which produces individually unique tests for each of the twelve program levels, and a set of six one-hour audio tapes. When used in conjunction with Latin Skills, these materials make it possible for teachers to supervise a self-paced program in elementary Latin (Culley, 1986).

Another source of some satisfaction has been the acceptance of generative CAI by other Latin software developers. To date there are no more than twenty commercial Latin instructional programs, most of them simple, hard-coded drill-and-practice; but in the last two years, Latin Skills has been joined by at least three other programs which employ the generative approach. Others seem to agree that it helps.

So much for the benefits that generative CAI provides. One can hardly spend this long with a technique without seeing its disadvantages. Even if you could manage to avoid seeing them, colleagues will be glad to point them out.

These routines are designed to produce all of the forms of the Latin verb, noun, adjective, or pronoun--the parts of speech which undergo change of form. The trouble is, many Latin words occur in only some of the forms they could theoretically assume. For example, **caelum**, sky, never occurs in the plural in all of classical Latin literature. Yet these programs, if ordered to do so by the student, will happily display plural forms for the word. Or take **fulgeo**, to gleam. This verb is mostly limited to the third personal forms like **fulget**, it gleams; but a student parsing forms and making errors can produce forms like **fulgemur**, we are gleamed. One of the chief strengths of the generative approach, its ability to produce forms dynamically in response to student input and use those forms as corrective feedback, turns out to be downright dangerous.

For all the flexibility of format and variety in application, the generative approach is simply a means of generating or recognizing different forms. If the pedagogical goal is something other than morphology, generative CAI can be stretched and bent to fit, but it feels the strain.

The reason for both of these problems, is, of course, that generative CAI knows nothing of meaning. Any appearance of such knowledge is only fancy footwork. And sooner or later that shortcoming makes itself felt. On occasion I have called lessons which use this approach "smart". That is their trouble: they are only "smart" when they need to be intelligent.

Ten years with the Latin Skills project have made it clear to me that the generative approach, powerful as it is for many applications, is only a stage in the development of foreign language CAI. Language, after all, is about the communication of meaning. Programs that cannot handle semantic content will ultimately be inadequate. And so, with still more of my University's generous support, I've spent a good part of the last three years testing the waters of artificial intelligence. Perhaps in another seven years, if we are all still around, I can tell you whether that worked.

References

Culley, G. R. (1984). The Delaware Latin skills project. *The Classical Outlook*, *62*, 38-42.

Culley, G. R. (1984-1986). *Latin Skills* [Computer program]. Newark, DE: Office of Computer-Based Instruction, University of Delaware. (Five-diskette series for Apple II+, //e and IIc microcomputers)
  Version keyed to Frederic M. Wheelock's *Latin: An Introductory Course*, 3rd ed. (Harper and Row, 1963) published 1984.
  Version keyed to Ullman, *Latin for Americans*, *First Book* and *Second Book* (Macmillan Publishing, 1981) published 1984.
  Version keyed to Jenney & Scudder, *First Year Latin* and *Second Year Latin* (Allyn & Bacon, 1984) published 1985.
  Version keyed to Goldman and Nyenhuis, *Latin Via Ovid*, 2nd ed. (Wayne State Univ. Press, 1977) published 1985.
  Version keyed to *Ecce Romani*, North American Edition (Longman, Inc., 1985, 1986) published 1986.

Culley, G. R. (1986). *Individualized Latin Curriculum*. [Self-paced program consisting of syllabus, test-generating diskette (for Apple II) and six audio tapes to accompany *Latin Skills*] Newark, DE: Office of Computer-Based Instruction, University of Delaware.

Culley, G. R. (1987). *Lector*. [Computer program]. Newark, DE: Office of Instructional Technology, University of Delaware. (For Apple II+, //e and IIc microcomputers)

Decker, H. W. (1976). Computer-aided instruction in French syntax. *Modern Language Journal*, *60*, 263-267.

Lippman, D. (1982). Computo, computas, computat. *Microcomputing*, (March, 1982), 136-139.

Muller, M. P. (1975). *COLA* [Computer program]. Paris: I.N.R.D.P. Section Informatique et Enseignement.

Olmsted, H. M. (1975). Two models of computer-based drill: Teaching Russian with APL. *Slavic and East European Journal*, *19*, 11-29.

Phillips, R. (1981). *Practicando Espa§ol con la manzana II, part I (verb drill) and part 2 (vocabulary drill)* [Computer program]. Iowa City, IA: CONDUIT. (Diskette for Apple II, 48K)

Weller, H. P. (1976). An experiment in computer-assisted instruction in elementary Spanish. *Proceedings of the 1976 Conference on Computers in the Undergraduate Curriculum* (SUNY-Binghamton, 1976), 281-28.

Gerald R. Culley
Department of Languages and Literature
University of Delaware
Newark, DE 19716
(302) 451-2596

INTEGRATING PROBLEM SOLVING AND TUTORIAL STRATEGIES:

EXTENDING THE GENETIC GRAPH

Garland Bayley, University of Portland

William S. Bregar, Tektronix Inc.

Arthur M. Farley, University of Oregon

ABSTRACT

The focus of this paper is on the underlying knowledge base for an intelligent tutorial system for high school algebra problems. We present a model of problem solving consisting of separate rule bases for strategic and computational knowledge. Within these knowledge bases, rules are related using the genetic graph (Goldstein, 1979), extending it to account for several new relationships. We then show how our representation can be employed by a tutorial system to decide what a student is doing and what kind of advice should be presented, when warranted.

INTRODUCTION

An important function of any intelligent tutorial system is its ability to monitor student behavior and react appropriately to student performance. Underlying this function there must be a model which can account for students at different levels of ability. Ability can be inferred from student problem solving behavior.
Among the indicators are

- the solution path a student might follow
- discernible strategies a student employs
- errors a student makes.

We present in this paper a problem solver which is designed to serve as such an underlying knowledge base for an intelligent tutor for high school algebra problems. It derives its flexibility through the separation of knowledge components into more general strategic knowledge and problem specific computation knowledge (Bregar, Farley, and Bayley, 1986; Clancey, 1984). These can be tied together, however, through an extended version of a genetic graph structure (Goldstein, 1979) which allows the system to easily move between strategies, represent erroneous student actions in the form of deviations and provide a basis for explanation of the origin of rules with respect to the student's perceived knowledge.

THE PROBLEM SOLVER

The problem solver solves problems by employing strategies to select computation rules. The strategies reflect levels of expertise ranging from novice to expert and are represented by rule sets. We account for several strategies, three of which we refer to as weak forward-directed, means-ends, and expert. Weak forward-directed can be characterized as one in which rules are selected to generate new values whenever possible but without apparent goal-directed behavior. It nevertheless retains the capacity for recognizing when the problem has been solved. This strategy is typically associated with novices who have minimal domain knowledge and few, if any, problem solving skills. At the next level, means-ends (Newell & Simon, 1972) works in a goal-directed fashion to reduce differences between the problem state and the goal state. Finally, the expert strategy combines goal-directedness with deeper knowledge of the domain and the computations to immediately set up problems and solve them. Appendix A presents an English description of these three strategies.

The computation rules describe how

quantities can be computed. When fully instantiated they can be activated to compute quantities in question (when all variables are bound to constant values) or used to create equations which can subsequently be solved as the unknowns obtain values. These are employed in conjunction with a set of entity descriptors or structured data types that characterize objects typically found in a class of problems. Different problem types can be processed by interchanging the problem dependent computation rules and their associated entity descriptors with those for another problem type.

THE REPRESENTATION AND ITS USE

When interacting with a student we map each step the student makes into an action. Actions consist of the selection of strategic rules and operations upon computation rules to produce new states. Our representation depicts not only what strategic and computation rules have been selected, but how they relate to other strategic and computation rules in the system. For a given rule there may, for example, be a related rule that should be made explicit for tutoring purposes. The representation should depict these relationships. Thus, to tie our rules together and to represent their relationships, we employ a variant of the genetic graph (Goldstein, 1979). We relate our rules to show levels of strategic expertise and to provide a representation for incorrect or deviant rules which can be used to diagnose a student's erroneous behavior. We provide an additional set of relationships among the computation rules which tie variants of given rules to problem complexity (in the sense of how difficult a problem is for a student to solve), and we extend the genetic graph to provide more domain specific information.

Computation rules

Computation rules are special forms of equations that already have a single quantity isolated on the left-hand side referred to as the output. The quantities on the right-hand side are referred to as inputs and when they have numerical values the rule can fire and calculate the output.

Goldstein defines four relationships between rules, generalization/specialization, simplification/refinement, analogy, and deviation/correction. One rule is a refinement of another if the first is applicable in a subset of the situations to which the other applies. For example, one of our simplest computation rules is

> CR1: The amount of an entity is equal to the sum of the amounts of its parts.

We can define two refinements of the above general rule as follows:

> CR1a: The amount of a composite solution is the sum of the amounts of the two participating solutions.
>
> CR1b: The amount of a substance in the composite solution is the sum of the amounts of the like substances in the participating solutions.

Each of these refinements is reflected in the constraint slot of the rule. Note that CR1a and CR1b are, in fact, analogous. CR1b results from substituting solutions with substances in the structure slot and replacing the constraint slot in CR1a with that of CR1b.

As originally described the relational links of the genetic graph are based on general concepts independent of a particular domain. However, to support representation of problem solving in the algebra word problem domain, we have defined additional links for relating computation rules. Two of these we refer to as derivation and combination. In the derivation link we further refer to base rules and derived rules. A rule is said to be derived from a base rule if it binds its variables to the same entities as the base rule and the quantity it computes is an "input" to the base rule. Base rules reflect the most "familiar" form of a computation. That is, for example, it is more reasonable to think of the whole being the sum of the parts than to think of a part as being the whole minus the other part.

Combination forms a new rule from two or more existing rules. The right-hand side (computation part) of a rule is substituted into the right-hand side of another base rule for a quantity that matches the left-hand side (output part) of the first rule. If the substitutions are only made for inputs to a base rule then a new computation rule is formed that is more complex and computes the same output.

Evidence in our protocols indicates that a problem's difficulty can be tied to the extent to which it can be described in terms of derived and combined rules.

To illustrate these concepts, consider CR1b, above and another rule (not shown formally, here) that states "the amount of a substance in a solution is the proportion of that substance times the total amount of the solution." Call this rule CR2. Its output is amount(substance) and its computation is proportion(substance) x amount(solution). We can substitute this computation for any of the amounts in CR1b, including its output. The result is an equation of the general form

$$p_1A_1=p_2A_2+p_3A_3$$

where p_i and A_i are proportions and total amounts of solutions, respectively. Such an equation is exactly that which is used by experts in solving "difficult" mixture problems.

Strategic Rules

The strategic rules also have genetic relationships between them that can be exploited for tutoring purposes.

We start with the generalization/specialization relationship. When a strategic rule is instantiated the instantiation is specialized, and the original strategic rule is its generalization.

In our representation simplification/refinement links group the rules into sets of rules with like goals. An example of a simplification/refinement relationship is the relationship between strategic rule ME1 and rule EX1 (see Appendix A). ME1 can be used to propagate a result upward in the goal tree only in a subset of the situations where EX1 would apply.

As an example of analogy links consider the rules ME7 and ME8. An analogy can be drawn between them in that ME7 can be selected in a situation where there exists a trivially solvable system of equations with one unknown while ME8 is useful for a system of equations with two unknowns.

Finally, the deviation/correction relationship can be shown by the relationship between the rules ME1, FD1, and EX1, and a rule that characterizes the behavior of students who take a nongoal quantity and use it as the goal. Such a rule can be written as deviation rule DR1:

> If an attribute of an entity has just been assigned a new value, use it as the value of a goal attribute.

Note that this rule is related to more than one correct rule. The system, monitoring the student, knows which of the correct rules is the proper rule under the circumstances.

TOWARDS A TUTORIAL SYSTEM

Systems such as GUIDON (Clancey, 1979) work in probabalistic domains. Algebra word problems present a more concrete domain within which to analyze student behavior. Anderson (1985) has developed the model-tracing paradigm for such domains. We show how a tutorial component can follow this paradigm and employ information from our representation of problem solving to provide reasonable tutorial guidance. Our approach is similar in philosophy to Greeno's (1978)

> ... skill in learning involves development of skills at one level and then integration of those skills into higher level units

Thus, our tutorial issues include 1) when has a subskill been acquired, 2) which subskill(s) to work with next, 3) how subskills combine to form higher-level skills, and 4) when to undertake tutoring of higher-level skills. In the following discussion we use the following notation to refer to elements of mixture problems. Assuming that a mixture problem involves up to two solutions combined to form a third, and that each solution consists of at most two substances, we have

A_i = the total amount of solution$_i$, $i = 1,2$

a_{ij} = the amount of substance$_j$ in solution$_i$, $j = 1,2$

p_{ij} = the proportion of substance$_j$ in solution$_i$

After each student input, a problem solving state exists and is posted to a global memory. At any point at which the tutorial system would intervene, a decision on what to say is made by examining the strategic rules and addressing the tutorial issues cited above. Examination of the strategic rules starts with the conflict sets for each of the strategic rule sets at the current system state. For example in one of our student-expert protocols (not shown) we observed the following state. Given the problem

> A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% alcohol solution?

The student had

1) noted the goal of finding the amount, A_2, of pure alcohol,

2) established the relationship amount (70% solution) = amount (20% solution) + amount (pure alcohol) which can be written symbolically as

   (1) $A_3 = 10 + A_2$.

   (A_2 is the goal, and it is known that the amount of the 20% solution is 10 gallons.)

3) determined that the amount of 70% solution, A_3, should be established as a subgoal.

Several possibilities now arise from the conflict sets. The weak forward-directed strategy would suggest computing any of the unknown proportions in the problem or computing that there are 8 gallons of water in the 20% alcohol solution. (Note how this corresponds to a novice's typical behavior--computing whatever is possible with little goal direction.) Means-ends suggests three possible actions:

1. Set up the equation

$$.7(A_3) = 0.2(10) + 1.0(A_2)$$

which can be solved simultaneously with (1) above.

2. Generate a new subgoal from one of

$$a_{21} = 1.0(A_2)$$
$$a_{31} = .7(A_3)$$

3. Generate two new subgoals from one of

$$A_3 = a_{31} + a_{32}$$
$$A_2 = a_{21} + a_{22}$$
$$a_{32} = p_{32}(A_3)$$
$$a_{22} = p_{22}(A_2).$$

For example, in the third equation above, a_{32} and p_{32} are new unknowns which would be become new subgoals. Finally, the expert strategy suggests that the goal, A_2, can be solved with

$$2 + 1.0(A_2) = .7(10 + A_2)$$

and subgoal A_3 can be solved with

$$2 + 1.0(A_3 - 10) = .7(A_3).$$

In the protocol, the student requested help. The tutor has four options

1) reinforce the lower level strategy using known rules, and enlarge his computation rule set.

2) give the student practice at the current strategy with known rules,

3) expand horizontally, keeping the student at the same strategic level but enlarge the student's computation rule set.

4) upgrade the student's strategic repertoire by expanding vertically to a higher level strategy,

The problem in this example is typically considered difficult, and although it can be solved using a weak forward-directed strategy, such a strategy is not optimal. The student has shown some degree of proficiency with this strategy, and has also given evidence of applying means-ends. Hence, the tutor might expand horizontally assuming a strategy level of means-ends and suggest that the student utilize the equation

$$.7(A_3) = 2 + 1.0(A_2)$$

as shown above. The system can trace the derivation of this equation by noting that it reflects the notion that the amount of a substance in a composite is equal to the sum of the amounts of that substance in each of the two solutions comprising the composite. We can express this as

$$a_{31} = a_{11} + a_{21}$$

Since the problem solving system knows that $a_{31} = .7(A_3)$, $a_{11} = .2(10)$ or 2, and $a_{21} = A_2$ because it is pure alcohol, it can provide further explanation to the student on how elements of the equation were obtained.

SUMMARY

We have presented a system that is designed to serve as the underlying knowledge base for an ICAI system to teach algebra word problems at the high school level. We extend the idea of the genetic graph and apply it to a practical, solvable problem in ICAI. We have identified and implemented several problem solving strategies in our protocols, and our system can monitor behavior that mixes these strategies. We have implemented the representation described, the problem solver, and a model-tracing system that verifies correct behavior, explains some typical errors it observes, and provides rudimentary help. A more complete tutorial component of the system is still under development.

REFERENCES

Anderson, J. R., Boyle, C. F., and Yost, G. (1985) The Geometry Tutor. Proceedings of the Ninth International Joint Conference on Artificial Intelligence. Los Angeles, CA, August, 1985, 1-7.

Bregar, W. S., Farley, A. M., and Bayley, G. (1986) Knowledge sources for an intelligent algebra tutor. Computational Intelligence. 2, 117-129.

Clancey, W. J. (1979) Tutoring rules for guiding a case method dialogue. International Journal of Man-Machine Studies. 11, 25-49.

Clancey, W. J. (1984) Methodology for building an intelligent tutoring system. In Kintsch, W., Miller, J. R., and Polson, P. G. (Eds.), Method and Tactics in Cognitive Science. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Goldstein, I. P. (1979) The genetic graph: A representation for the evolution of procedural knowledge. International Journal of Man Machine Studies. 11, 51-77.

Greeno, J. G. (1978) Understanding and procedural knowledge in mathematics instruction. Educational Psychologist. 12, 262-283.

Newell, A., and Simon, H. A. (1972) Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall Inc.

APPENDIX A: STRATEGIC RULES

Means-ends rules (ME)

1. Given a goal attribute with a known value, note solution and utilize answer (report result or propagate upward in goal tree).
2. Given a goal attribute, select a rule computing its value as output and determine consistent entity bindings.
3. If all input attributes of a bound computation rule have known values, perform the computation rule.
4. Given a bound computation rule with unknown attribute value inputs, make the unknown input attributes goals.
5. Given a goal attribute, select a rule having it as input and determine consistent entity bindings.
6. Given a bound computation rule, activate the algebraic rules to create an equation corresponding to the rule.
7. Given a goal attribute and an equation with its corresponding variable, activate the algebraic rules to solve the equation for that variable (ie. isolate it on the left hand side).
8. Given two equations having a non-goal variable in common, activate the algebraic rules to combine equations, eliminating the common, non-goal variable.

Forward-Directed rules (FD)

1. If the goal attribute has a known value, note the solution and utilize the answer.
2. If all input attributes of a bound computation rule have known values, perform the computation rule.
3. Given an attribute with a known value, select a rule with it as input and determine consistent entity bindings.

Expert rules (EX)

1. Given a goal attribute with known value, note result and utilize answer.
2. Given a new goal attribute, select a known algebraic relation among it and the set of known attribute values and solve for the goal attribute with algebraic rules.
3. Given a goal attribute, employ means-ends rules to generate new goal attribute(s).

Garland J. Bayley
5000 North Willamette Blvd.
Portland, OR 97203-5798
(503) 283-7163

An Electronic Coaching System to Teach Concepts of Personal Causation and Symbol Use to Severely Handicapped Learners

Gaye V. Fedorchak, Washington State Department of Social & Health Services

Steven E. Nantz, Washington State Department of Social & Health Services

ABSTRACT

Designed to be testable and verifiable, this instructional system is enabling instructional researchers in Washington State to examine the developmental progression novices follow as they gain increasingly complex mastery of cause-effect and symbol-outcome relations. This presentation will be of interest to instructional researchers, special educators, and to those interested in building developmental models of the acquisition of expert problem solving skills.

PERSONAL CAUSATION & SYMBOL USE

In 1986, the Personal Causation Training Program was begun as a special project for Fircrest School, a division of the State Department of Social & Health Services, which serves handicapped persons. The Personal Causation Program is an integrated electronic coaching system designed to teach persons with severe and multiple handicapping conditions to actively master events in their daily living environments over which they have never before had control. Our goal, in effect, is to teach these individuals to independently solve increasingly difficult cause-effect problems to obtain enjoyable and reinforcing stimulation. Though project development is still underway, preliminary results are extremely encouraging. The outcome of this project, we believe, will be to show that profoundly retarded persons with multiple motoric limitations can in fact learn to master their environments in ways seldom demonstrated. In the process, this coaching system will enable us to identify and describe developmental patterns in the acquisition of the most fundamental problem solving skills: personal causation and symbol use.

This paper will describe the basic structure and cognitive task analysis which support the Personal Causation Coaching System. To achieve our instructional goals, we are developing coaching tools of two types:

The Electronic Tool

First, we are designing and building electronic switch systems adapted to the **functional motoric level** of the individuals involved. These individually tailored adaptive devices constitute a relatively new problem solving tool which has seldom been available to this often neglected population. Electronic switches are a tool which will allow persons to explore cause-effect relations in their environment which they have never before been able to explore.

From a management perspective, we are also developing 2 other systems: 1: A remote bar-code based data collection system which can be downloaded directly into our database (using the Videx Timewand System), and 2. Switch interfaced computer control programs for the Macintosh which will better allow presentation of cause-effect problems to learners at the higher difficulty levels. In sum, the "electronic tool" is designed to:

1. Match each person's functional motoric level, thus providing opportunities for environmental exploration otherwise not available.
2. Provide unobtrusive on-site monitoring of learning outcomes, and to
3. Control the problem structure and timing of reinforcement at the higher difficulty levels.

The Electronic Tool, however, is only one of the tools being developed by this project. The electronic systems alone are not enough to do the job of teaching persons with profound cognitive handicaps to solve many environmental mastery problems.
In fact, a much more fundamental tool is needed.

The Instructional Tool

The second, but more important tool being developed by this project is the Instructional Tool - our developmentally based training curriculum. This tool is an experimental coaching system for teaching increasingly difficult problem solving skills to persons previously thought to be incapable of acquiring such skills. The instructional tool, therefore, is the tool designed to guide the use of the electronic switch systems to match each person's **functional cognitive level** of problem solving skills, and to help that person explore and progress to higher cognitive levels of problem solving. The "instructional tool" (our coaching system) is designed to:

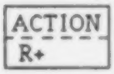
1. Assess an individual's initial level of cause-effect problem solving skill.
2. Provide a starting point for instructing the next higher problem solving skill, and
3. Provide a structure for teaching people to generalize their newly learned level of mastery to other settings and other reinforcing events, and to
4. Guide the design and development of electronic systems whichwill support optimal learning.

THE PERSONAL CAUSATION PROGRAM- A CONCEPTUAL OVERVIEW:

The first coaching task involves the developmentally-based assessment of each person's cause-effect problem solving behavior. Performance at 5 levels of increasingly difficult cause-effect problems is assessed in the following way: Given a switch system which matches the person's motoric capabilities (i.e. a simple push button switch); and an environmental event which has been shown to be preferred by the individual (i.e. a favorite tape recorded song), the switch and the reinforcer are presented to the individual in one of 5 different cause-effect conditions. These 5 problem solving conditions vary in difficulty from one condition that is very simple and immediately reinforcing, to a condition which is very difficult and would require the ability to delay gratification.

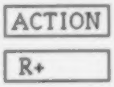
The five cause-effect levels assessed include the following conditions:

Level 1:



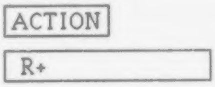
At Difficulty Level 1, both the action (a switch press) and the reinforcing outcome (music) occur in the same place and at the same time. The switch is placed "on top of" the reinforcer, and from the moment of press to the moment of release the reinforcement continues. Thus, the action and the outcome occur in exactly the sameplace and for exactly the same time, and simple cognitive association between action and outcome is formed.

Level 2:



At Difficulty Level 2, the action and the reinforcing outcome still occur for exactly the same time, but now the press and the outcome are separated in place. The switch is moved away from the reinforcer. As soon as the person presses, however, the music will still play for exactly the same time as the press occurs, and will stop at the moment the press is released. The increased difficulty comes, according to our theory, from the change in one feature only - the separation of action from outcome in space. The cognitive association between action and outcome is now a little less simple.

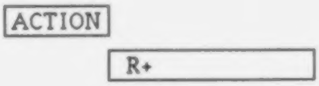
Level 3:



At Difficulty Level 3, one more feature is changed in the way action and outcome are related. At this problem solving level, just like at the previous level, the switch press and the reinforcer start at the same time, and are separated in place. One feature is different, however. When the switch press is released, the reinforcer continues for some time after, despite the fact that the switch is no longer being pressed. Thus, the time of offset between action and outcome is now different. The association between this action and this outcome is now a bit more difficult.

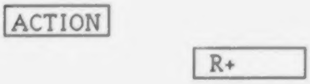
Literally, the outcome is further removed from its cause (the person's action) in two features from Level 1. Now, they are separated both in place and time of offset.

Level 4:



At Difficulty Level 4, again, one more feature is changed in the relation of action to outcome. Now, as before, the switch press and outcome are separated in place, and they still partially overlap in time, and the reinforcer continues for some time after the release of the switch, but one more feature has been changed. Now, although their times still overlap at a single point, the time of onset of the reinforcer is delayed relative to the onset of the action. Now, cessation of action may be required for reinforcement to occur. The press occurs, then a pause, then the press is released, the reinforcment starts, and a little while later, the reinforcement stops. The action and outcome are now completely separated in place, and almost completely separated in time, except that they are tangent at one point in time. The problem of what causes the reinforcement is becoming more difficult as the association between action and outcome is pulled increasingly apart in place and time.

Level 5:



At Difficulty Level 5, the most difficult level assessed in the current system, the switch press and reinforcing outcome are completely separated in both place and time. There is now no overlap between the action and outcome at all. In fact, this cause-effect condition is an example of what has come to be known as delayed gratification, a behavior that involves acting now to obtain a reinforcing outcome at another time, and at another place. It is very similar in concept to many token reinforcement training systems often used with higher functioning persons. In fact, if a profoundly handicapped person were to reach this training level, a token reinforcement system might be introduced at this point. Hence, we have just outlined a possible training bridge between profound mental retardation and more moderate levels of mental retardation.

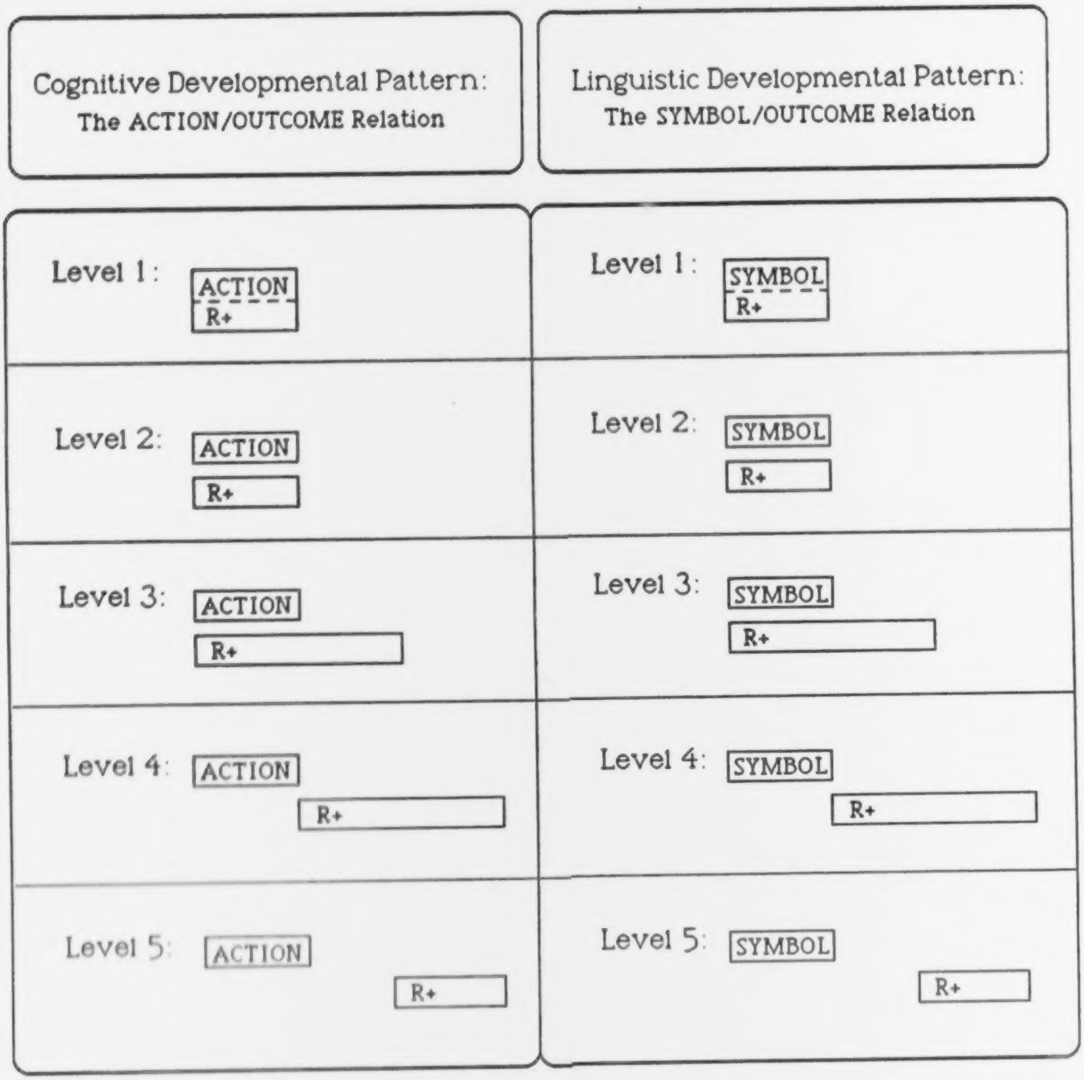
Moreover, it is also possible to consider this progression of increasingly complex knowledge levels as a guide for the instruction of basic symbol-outcome relations, or early symbol use. As a person mastered the first three action/outcome and symbol/outcome association knowledge levels, they would be demonstrating increasing readiness for the introduction of simple communication board training. Use of a communication board may be very similar cognitively to cause-effect knowledge levels 4 and 5, both of which require a person to act now (i.e. indicate a symbol) to gain reinforcement later (obtain a response from another person). For a graphic comparison of the cognitive action/outcome relation and the linguistic symbol/outcome relation, see Appendix A.

Once reinforcers have been identified and motoric requirements have been met with development of an appropriately positioned switch system, the procedure for conducting the initial Baseline Assessment employs graduated guidance prompting with a backward fading technique to allow systematic observation of the highest knowledge level at which the individual shows independent mastery (criterion = 80% independent activation of reinforcement over trials at any knowledge level).

The formal Acquisition Training program is started for the individual at the lowest knowledge level where fewer than 80% independent activations have been shown. During Acquisition Training a normal (most to least) prompt fading technique is employed to encourage increasingly independent activation at that difficulty level. In effect, the prompting system used creates a guided discovery approach to the learning process. With appropriately designed switches, throughout the program each person is offered repeated opportunities to accidently or purposefully explore the problem. This process enables the student to independently discover and master increasingly complex cause and effect relationships. Systematically faded graduated guidance prompts are used only for those persons who demonstrate completely passive, nonexploratory behavior.

In addition, a Generalization Training program has also been developed and is derived from the Acquisition Training procedure. The intent of the Generalization program is to enable the individual to generalize their initial mastery of one reinforcing event to other reinforcing events and other settings at the same cognitive difficulty level.

APPENDIX A


Cognitive Developmental Pattern:
The ACTION/OUTCOME Relation
Linguistic Developmental Pattern:
The SYMBOL/OUTCOME Relation
Level 1:
ACTION
R+
Level 1:
SYMBOL
R+
Level 2:
ACTION
R+
Level 2:
SYMBOL
R+
Level 3:
ACTION
R+
Level 3:
SYMBOL
R+
Level 4:
ACTION
R+
Level 4:
SYMBOL
R+
Level 5:
ACTION
R+
Level 5:
SYMBOL
R+

SUMMARY

With the development of these instructional tools, it is clear we are breaking new ground in this instructional project. In fact, both our electronic and instructional tools can be considered to exploratory and experimental in their present form. Currently we are also developing a second instructional tool, which will be specifically designed to coach the acquisition of increasingly complex chains of behaviors.

In addition to the development of new direct training services for this population, a primary purpose of this project is to collect solid data enabling us to more clearly identify sequential patterns in the development of simple cause-effect problem solving skills. Thus, establishing and refining and empirical basis for teaching environmental mastery skills is one important component of the development of this curriculum. We are developing a testable, verifiable and fully integrated electronic instructional system which, hopefully, will benefit other programs attempting to coach similar learners to function more independently.

Selected Bibliography

Butterfield, E.C. (1981) Instructional techniques that produce generalized improvements in cognition. In P.E. Mittler (Ed.), *Frontiers of knowledge in mental retardation: Prodeedings of the 5th congress of IASSMD*, (Vol.1), Baltimore:University Park Press.

Butterfield, E.C.; Siladi, D.; & Belmont, J.M. (1980) Validating theories of intelligence. In H. Reese & L.P. Lipsitt, (Eds.) *Advances in child development and child behavior*. (Vol.15). New York: Academic Press.

Fedorchak, G.V. (1985) A developmentally based coaching system designed to teach the self management of general problem solving. *Masters Thesis*. State Univer- sity of *New York at Oswego*. (Conducted at the University of Washington, Seattle, WA.)

Inhelder, B.; & Piaget, J. (1958) *The growth of logical thinking from childhood to adolescence*. New York:Basic Books.

Malone, T.W. (1984) Toward a theory of intrinsically motivating instruction. In D.F. Walker & R.D. Hess (Eds.) *Instructional software: Principles and perspectives for design and use*. Belmont:Wadsworth.

Seigler, R.S. (1981) Developmental sequences within and between concepts. *Mongraphs of the Society for Research in Child Development*. *46*, (2, Serial No.189).

Stokes, T.R. & Baer, D.M. (1977) An implicit technology of generalization. *Journal of Applied Behavior Analysis*, *10*, 349-367.

Videx TimeWand Systems and Software, c/o Videx, Inc., 1105 NE Circle Blvd., Corvallis, OR. 97330.

Inquiries should be directed to:

Gaye V. Fedorchak,
Instructional Psychologist
Department of Social & Health Services
15230 15th Ave. NE
Seattle, Washington 98155

A COGNITIVE METHOD FOR DESIGNING

COMPUTER BASED TRAINING

Veli-Pekka Lifländer, M.Sc.(Eng.)

Helsinki School of Economics and
Business Administration
FINLAND

ABSTRACT

Helsinki School of Economics and Business Administration started in april 1986 a two-year project to introduce computer aided instruction to its everyday training activities.

The main goals of the project are to build an advanced CAI laboratory for CAI production, to develop design methods and tools to be used in the laboratory, and give teachers and other experts competence to use these methods and tools.

In the last few years the design of CAI programs has been moving from behavioristic to cognitive theories of learning. Cognitive theories have been discussed a couple of years in CAI papers and conferences. Bruner (1966), Ausubel (1978) and Gagné (1965,1979) are frequently used references. Yet the educational psychologists who are familiar with more advanced theories of orientation and learning process have not been involved with CAI projects.

This paper presents a new approach to CAI design, where the bridge is build between good instructional design and its realization on computers. The outcome is an open learning environment that supports expansive learning.

Authoring tools for CAI have developed very quickly and new user interfaces brought with microcomputers have lowered tresholds for users that are not computer specialists. Graphics, animation, simulation, interaction, data bases and knowledge bases can be used in most advanced CAI-environments.

The hardware used with CAI is now typically IBM PC or Apple Macintosh type microcomputers. New input and output devices like mouses, video and voice digitizers and synthetizers, video discs and CD ROM devices have extented the scope of learning materials available in CAI.

New generations of microcomputers, one branch offering multitasking and windowing and the other, so called 5th generation computers offering intelligent knowledge based systems, are today too expensive to be used in other than very advanced training purposes. But tomorrow ...?

1 BASIC MODES OF COMPUTER AIDED INSTRUCTION (CAI)

1.1 Separate CAI programs

So far the largest group of CAI-programs are the separate programs used for teaching or demonstrating some special narrow subject. These programs may be simulations, models or even games and the subject matter may be e.g.some chemical, physical or economic process.

Separate CAI-programs are mostly used in classroom situations, where the teacher demonstrates the program to the class.

1.2 Computer based tutorials

Computer based tutorials are self-contained CAI-courses that teach a total subject matter and have typically tens or hundreads of screens and interactions.

The students may have several different ways to go through the training program, but these routes are principally designed by educational and content experts.

Tutorials are used by studends alone or in groups but mainly without a teacher. The subject matters are usually well-structured and have long training tradition.

Courses on computer literacy, use of computer software and business administration are typical subjects for CAI-tutorials.

1.3 Computer based learning environments

Computer based learning environment means an approach to learning different from CAI-tutorials. The student acts like researcher while working with the subject matter. The learning process has not been built in the evironment. The point is that the student has good tools to act with the subject and living information available in many different forms and views.

Typical subject matters might be management problems, environmental studies and many others where the knowledge is not yet well structured.

2 DESIGN METHODS OF CAI

2.1 Behavioristic versus cognitive approach

Computer aided instruction was at the beginning a domain of computer specialists and very few educational scientists. Behavioristic view of learning offered a simple and attractive approach with its clear and measurable learning objectives.

Splitting the subject matter into small pieces, rote learning and quick feedback were common features of behavioristic learning programs.

Older CAI-programs have been criticized to be boring and underestimating the student. One reason may be the poor quality of graphics and interaction. But more important is probably the bad design of learning process and the structuring of the subject matter.

Cognitive views of learning have been introduced in CAI-conferences and papers with critics against old programs and the behavioristic views. (See e.g. Merrill 1987, Black 1987, Hannafin 1986.)

Critics has not yet widely led into new practices. The main reason is that the critics has come from educational scientists and the most people working with CAI have learned strictly behavioristic ideas of instructional design.

Also the cognitive view applied seems not to be very advanced in orientation strategies and learning process. The orientation is discussed in terms of advance organizers (see e.g Ausubel & al 1978 and Bruner 1966). Gagne (1965,1979) is the most referred on his ideas about the conditions of a good learning process.

I will give some references to design methods which are on the move to cognitive approach: Barker & Yeates (1985 p.53), Jay T.B. (1983), Gagne,Wager & Rojas (1981), Allessi & Trollip (1985), Crossley & Green (1985), Smith & Garton (1986), De Diana et al.(1987), and Lifländer (1987)

2.2 Subject matter, orientation, learning process and a CAI program

The instructional design method introduced here is based on works of Yrjö Engeström (1985,1987) and some other finnish educational psychologists. The model of the learning process by Engeström is described in appendix 1. The roots of the approach can be found in the works of L.S. Vygotsky (SU)(and later e.g. A.N. Leontjev, P.J. Galperin and V. Davydov), D. Ausubel, J.S. Bruner (USA) and L.Klingberg (DDR) etc.

The main problem of the CAI-design method has been to build the bridge between instructional design and its realization to a computer program. The design process and tools are briefly described in appendix 2/1.

The first two phases of CAI design are similar to usual instructional design but more exact. The general part of the curriculum (appendix 2/2) contains general conditions of learning: structuring the subject matter and finding out the ZPD (zone of proximal development) for the students knowledge and working practices.

The second phase is described in appendix 2/3. It contains a very thorough design of the learning process. The design sheet is in appendix 3.

In phase 3, logical design of a CAI program (appendix 2/4), we first describe the elements of the learning environment.(See 'market design' by Crossley and Green 1985). Then we try to create clear, visual and flexible 'traffic rules' for the program and a prototype structure which enables further refinement and development of the subject matter and the learning process. The main tool in this phase is a diagonal matrix described in appendix 4.

Phase 4, visual design, is described in appendix 2/5. Its goal is to give to the program an interesting and appealing visual outlook that supports the subject matter and the learning process.

3 DESIGN AND CONSRUCTION TOOLS OF CAI PROGRAMS

3.1 Programming languages

The most flexible tool for CAI-programs is a programming language like C or Pascal. Many experienced authors have built a lot of procedures to produce text, graphics, animation, answer judging etc. Yet programming is not easy, and problems arise, while experts of teaching and subject matter try to tell the programmer, what they really want to do.

3.2 Authoring languages

Authoring languages are CAI-tools that are specially designed to transform pedagogical ideas into a CAI-course. Work with an authoring language is similar to programming, but many ready made special functions like screen design and answer judging make the process from design to realization interactive and quick. PILOT and TenCORE are well-known authoring languages.

3.3 Authoring systems

Authoring systems are menu-driven design programs. The author does not need programming skills but can focus on the subject matter and the learning process.

ADROIT is an example of an authoring system. TenCORE has also many features that make it an authoring system , specially the new menu-driven version to be introduced in 1987.

3.4 Other tools

Authoring systems may also have build-in tools for instructional design. Good examples of such systems are EDUCA (see De Diana et al. 1987) and Course of Action by Authorware Inc.

New windowing software that makes it easy to handle different information elements on screen windows (GEM, Framework, Microsoft Windows, Macintosh interface etc.) gives new tools for CAI-designers. Especially new operating systems for micros with multitasking (like UNIX and future versions of MS DOS) make windowing more flexible.

4 DEVELOPMENT OF CAI HARDWARE

4.1 New microcomputers

Computer aided instruction cannot be an economic way of teaching if the hardware needed is expensive and difficult to use. Fortunately the mass production of microcomputers has brought the prices down. Their wide use at work and home has forced the software companies to develop very easy and convenient user-interfaces with icons, mouses, touch screens etc.

IBM PC-compatible microcomputers are mostly used for new production of CAI-software. The new micros with built-in iconic interfaces (Macintosh, Atari, Amiga) have more advanced input and output deviced which can be used very effectively e.g. in music training. More advanced versions of these so called 32-bit supermicros (SUN, Apollo, Macintosh II) are mostly used for computer aided design and are so far too expensive for general training purposes.

The 5th generation of microcomputers with parallel processing make it easier to build expert systems as a part of the learning environment.

This approach has been used in very high level expert education e.g. in medicine.

4.2 Laser discs

Laser discs with optical analog or digital coding systems are promising devices for information retrieval in CAI.

Videodiscs (Laservision) can store 54000 addressable still pictures or 1 hour living video or 75 hours digital voice or 884 megabytes of computer data or any combination of these. The most advanced training applications so far have been made in music, medicine, technical and military training.

CD ROM (Compact disk read only memory) is using same techniques as CD music discs. The standard introduced by Philips and Sony can store 550 megabytes of digital data. The data can represent also voice and graphics.

A promising application of CD ROM-techniques have been different thesauri and large databases.

5 CONCLUSIONS

5.1 CAI-applications

The most promising field of CAI seems to be a learning environment where very large domain of knowledge and information in different forms can be studied and handled by the students using all the power of the computer in this work.

A very effective way of using the CAI-materials is to use it in small groups. So the learning situation may be much more living and motivating.

5.2 Methods and tools of CAI-design

In designing CAI-courseware good expertise of the subject matter and the instructional design is needed. Good design of orientation and the whole learning cycle (motivation, orientation, presentation of knowledge, fixing the knowledge, application and evaluation) must precede the realization of the training process on computer.

The CAI-process must be well guided but at the same time very flexible from the student's point of view.

Best tools for authoring CAI-courses are based on new visual user interfaces and result on learning environments, where the students can really work and act with the knowledge.

5.3 Hardware

Visually oriented microcomputers, training programs and laser discs are technologies which can be duplicated with relatively low costs. This means that in mass production the unit prices get remarkably low.

REFERENCES

Allessi Stephen M. & Trollip Stanley R. (1985)
Computer-Based Instruction: Methods and Development
Prentice Hall, Englewood Cliffs, New Jersey 1985

Ausubel D.P., Novak, J.D.,Hanesian H. (1978)
Educational Psychology: A cognitive view
New York: Holt, Rinehart & Winston

Barker Philip & Yeates Harry (1985)
Introducing computer assisted learning
Prentice Hall international 1985

Black Thomas R. (1987)
Extending the use of authoring systems
In Moonen and Plomp (Eds.):eurit 86, Developments in Educational Software and CoursewareEuropean conference on information technology
Pergamon Press 1987

Bruner Jerome S. (1966)
Toward the theory of instruction
Cambridge, Harvard University Press 1966

Crossley K. & Green L. (1985)
Designing Computer Lessonware:
A Practical Guide for Teachers
ISBN 0-9692382-0-7
Crossley & Green, Ontario, Canada 1985

De Diana I.,Bulthuis W. & Van Schaik P. (1986)
Design methodologies in computer aided learning.
Materials of a pre-conference seminar
eurit 86, Twente University of Tecnology
The Netherlands 20.5.1986

Engeström Y. and Hedegaard M. (1985)
Teaching theoretical thinking in primary school: The use of models in history/biology
In E. Bol, J.P.P. Haenen & M.A. Wolters (Eds.) Education for cognitive development.
Den Haag:SVO/SOO (p.170-193)

Engeström Yrjö (1987)
Learning by Expanding - An Activity-theoretical Approach to Developmental research (368 pages)
ISBN 951-95933-2-2
Gummerus 1987, Jyväskylä,Finland

Gagné R.M and Briggs L.J. (1979)
Principles of instructional design
Holt, Rinehart and Winston, New York 1979

Gagné R.M (1965)
Conditions of learning
Holt, Rinehart and Winston, New York 1965

Gagné R.M., Wager W. & Rojas A. (1981)
Planning and authoring computer-assisted instruction lessons
Educational Technology September 1981, pp.17-21

Hannafin M.J. & al. (1986)
The effects of orienting strategies and cognitive processing time on factual and inferential learning
ADCIS 27th conference proceedings pp. 190-194
Association for development of CBI-systems
Washington 1986

Jay T.B. (1983)
The cognitive approach to computer courseware design and evaluation
Educational Technology, January 1983, pp. 22-26

Lifländer Veli-Pekka (1987)
A Method for designing computer aided learning environments
Nordisk konferens för datostödd utbildning i universitets- och högskoleutbilningen (pp. 55-71)
(A Scandinavian conference of computer based education in universities)
Reprocentral I Kungliga Tekniska Högskolan
100 44 Stockholm
ISBN 91-7170-906-1

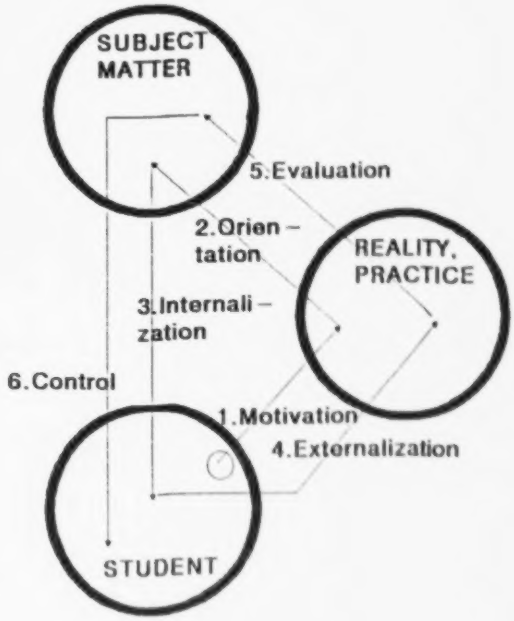
Merrill M. David (1987)
Keynote address: Component Design Theory: Instructional design for Courseware Authoring
In Moonen and Plomp (Eds.):eurit 86, Developments in Educational Software and CoursewareEuropean conference on information technology
Pergamon Press 1987

Smith Lynn & Garton Roland (1986)
Better chemistry through cai - getting started
ADCIS 27th conference proceedings pp. 221-226
Washington 1986

Veli-Pekka Lifländer
Helsinki School of Economics
and Business Administration
Runeberginkatu 14-16
SF-00100 HELSINKI
FINLAND
Phone: 358-0-4313633

Appendix 1

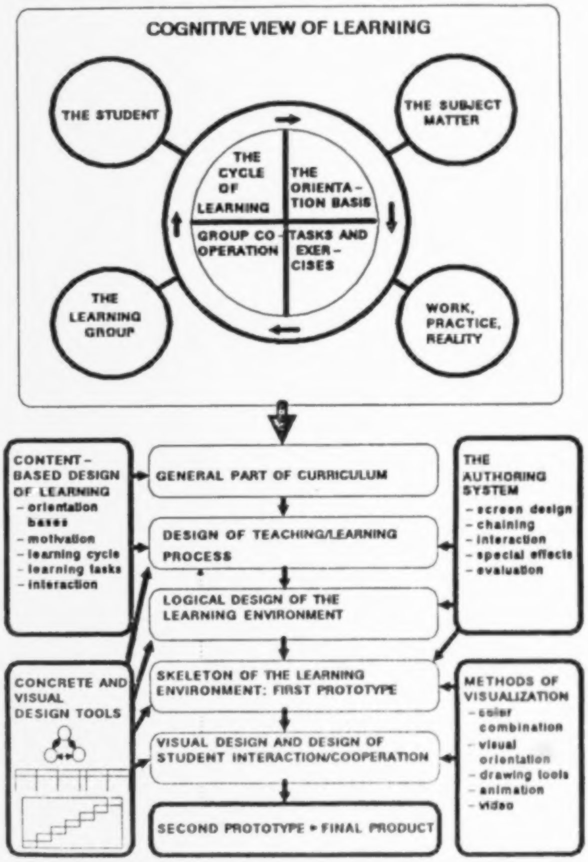
A MODEL OF THE LEARNING PROCESS (BY YRJÖ ENGESTRÖM)


SUBJECT MATTER
5. Evaluation
2. Orien-tation
REALITY, PRACTICE
3. Internali-zation
6. Control
1. Motivation
4. Externalization
STUDENT

Veli - Pekka Lifländer

Appendix 2/1

DESIGNING COMPUTER AIDED LEARNING ENVIRONMENTS


COGNITIVE VIEW OF LEARNING
THE STUDENT
THE SUBJECT MATTER
THE CYCLE OF LEARNING
THE ORIENTA-TION BASIS
GROUP CO-OPERATION
TASKS AND EXER-CISES
THE LEARNING GROUP
WORK, PRACTICE, REALITY
CONTENT-BASED DESIGN OF LEARNING
- orientation bases
- motivation
- learning cycle
- learning tasks
- interaction
GENERAL PART OF CURRICULUM
DESIGN OF TEACHING/LEARNING PROCESS
LOGICAL DESIGN OF THE LEARNING ENVIRONMENT
SKELETON OF THE LEARNING ENVIRONMENT: FIRST PROTOTYPE
VISUAL DESIGN AND DESIGN OF STUDENT INTERACTION/COOPERATION
SECOND PROTOTYPE = FINAL PRODUCT
THE AUTHORING SYSTEM
- screen design
- chaining
- interaction
- special effects
- evaluation
CONCRETE AND VISUAL DESIGN TOOLS
METHODS OF VISUALIZATION
- color combination
- visual orientation
- drawing tools
- animation
- video

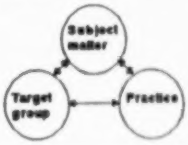
Veli - Pekka Lifländer **Appendix 2/2**

DESIGNING COMPUTER AIDED LEARNING ENVIRONMENTS

Phase 1: GENERAL PART OF THE CURRICULUM

Goal: To collect the background information for instructional design

Result:



Subtasks:

DEFINING THE TARGET GROUP

↓

QUALIFICATIONS PRODUCED BY TRAINING

↓

USES OF THE SUBJECT MATTER IN LIFE / AT WORK

↓

CRITERIA FOR SELECTING THE SUBJECT MATTER

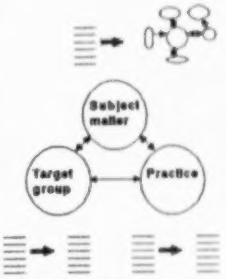
↓

CONTENT ANALYSIS AND THE PRIMARY ORIENTATION BASIS

↓

ECONOMIC AND ORGANIZATIONAL FRAMEWORK

Tools and methods:



Zone of proximal development in knowledge and working practices of the target group

---

Veli - Pekka Lifländer **Appendix 2/3**

DESIGNING COMPUTER AIDED LEARNING ENVIRONMENTS

Phase 2: DESIGN OF TEACHING/LEARNING PROCESS

Goal: Refined description of the teaching/learning process

Result:

See the design sheet in appendix 3

Subtasks:

Refined content analysis
- chunks of content and key concepts
- development trends and contradictions
- orientation bases

↓

Didactic phases and their content ideas (MO, OR, KR, IN, EX, EV, CO)

↓

Tasks to be introduced to the student

↓

Interaction of the learning group. Interaction between the student and the program

↓

The ways of controlling the learning results

↓

Scheduling the learning process

Tools and methods:

Expert discourse
Contradictory roles
Development history and original core

Design worksheet: see appx 3.
A3 – sheets with Post – It or computer screen sheets with paint or draw – programs (like GEM Draw or MacPaint)

Veli - Pekka Lifländer

Appendix 2/4

DESIGNING COMPUTER AIDED LEARNING ENVIRONMENTS

Phase 3: DESIGNING THE LOGIC OF A CAI PROGRAM

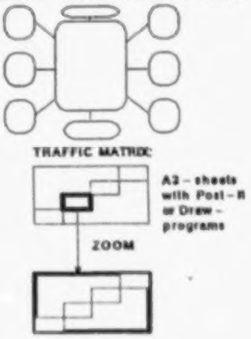
Goal: To create clear, visual and flexible 'traffic rules' for the learning environment. To create a prototype structure which enables further refinement and development of the subject matter and the learning process.

Result: A marketplace describing the structure of the learning environment. Program blocks with standard and situational ways of moving. Menus and instructions. Compound screens. Variable names and definitions.

Subtasks:

- Finding the components of the learning environmnet ('marketplace')
- Transforming the curriculum into lessons and units of a CAI - program
- Traffic rules of the learning environment
  - general rules (NEXT,BACK,INDEX etc.)
  - rules depending on choices
  - rules depending on learning results
  - menus,instructions and help functions
- Construction of the traffic map
- Construction of the advising system
- Construction of the menus
- Principles of the stude[illegible]ation

Tools and methods:



Visual menus and orientation bases as menus

Built - in or separate student administration systems

Veli - Pekka Lifländer

Appendix 2/5

DESIGNING COMPUTER AIDED LEARNING ENVIRONMENTS

Phase 4: VISUAL DESIGN OF A CAI PROGRAM

Goal: To create an interesting and appealing outlook that supports the subject matter and the learning process

Result: Color combinations, ways of expressing the didactic phase, visualized orientation bases, guidance, animations, title screens, 'rest screens' etc.

Subtasks:	Tools and methods:
Designing the color combinations	Color demonstrations of the authoring system
Visual design of orientation bases and other pictures	Paint or Draw - programs or the graphicsd editor of the authoring system
Visual design of the guidance system	Diagonal matrix used in logics design
Design of title screens and 'rest screens'	Paint programs
Design of animations	Pascal animation tools? Animation facilities of the authoring system
Design of other visuals (e.g. video)	Laser disc techniques of the authoring system

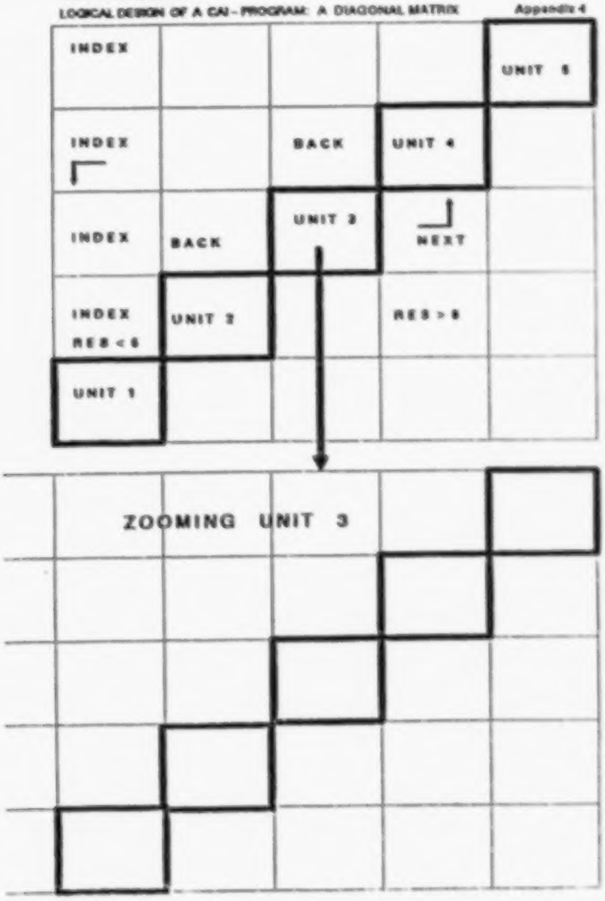
Veli-Pekka Lifländer 6.12.1986 **Appendix 3**

DESIGN OF THE LEARNING PROCESS

SUBJECT MATTER AND DEVELOPMENT PHASES OF THE GENERAL ORIENTATION BASIS	TEACHING UNITS, KEY CONCEPTS AND SPECIAL ORIENTATION BASES	DIDACTIC PHASE: MOtivation, ORientation, Knowl. Repr. INternaliz., EXternaliz., EValuation, COntrol AND ITS CONTENT IDEA	TIME	LEARNING TASKS	INTERACTION OF THE LEARNING GROUP

Veli-Pekka Lifländer

LOGICAL DESIGN OF A CAI-PROGRAM: A DIAGONAL MATRIX Appendix 4


INDEX
UNIT 5
INDEX
BACK
UNIT 4
INDEX
BACK
UNIT 3
NEXT
INDEX
RES < 5
UNIT 2
RES > 5
UNIT 1
ZOOMING UNIT 3

PRINCIPLES OF FEEDBACK AND RELATED VARIABLES IN COMPUTER-ASSISTED INSTRUCTION

Joseph S. Mattoon - Arizona State University

Division of Psychology in Education

ABSTRACT

A review of empirical research examines the effects of related variables of feedback for multiple-choice questions. Variables include frequency: how often to give feedback, content: what kind of information should feedback contain, occasion: when to present feedback, and learner expectancy: the learner's expectation of assessment and the level of confidence they have in their answers. Controlling these variables to optimize feedback is most likely to succeed through computer-assisted instruction.

Feedback (FB) to the learner is a major component of almost all instructional settings. The broadest definition of FB in instruction encompasses all communication made to the learner as a function of the learner's response to the instruction. This review is restricted to a discussion of empirical data relevant to the use of multiple-choice questions (MCs) with FB. FB refers to knowledge of correct response and consists of repeating the question stem, distractors, and the underlined target (correct choice). Variables that determine the effectiveness of FB are often neglected or poorly understood by instructional designers and educators. FB variables include frequency, content, occasion (time that FB is presented), and learner expectancy. It is often difficult or even impossible to adjust the delivery of a lesson as a function of these variables in standard, group instruction. Improvement of lesson delivery through analyzing student responses and utilizing the influence of FB variables may best be accomplished through computer-assisted instruction (CAI) (Kearsley, Hunter, & Seidel, 1983; Park & Tennyson, 1986).

Frequency of Feedback

A common misunderstanding of the role of FB has led many to believe that it is equivalent to "positive reinforcement" which increases the probability of repeating a response (Surber & Anderson, 1975). Such an hypothesis would lead one to believe that students would exhibit behavior that follows the principles of variable schedules of reinforcement (Skinner, 1981). Theories of reinforcement are based on the law of empirical effect; 'a learnable response followed by a reinforcing event, a stimulus state of affairs, will receive increment in its strength of a probability of occurrence' (Thorndike, 1981). These principles are well established and supported through behavioral research done with animals. If the principles of reinforcement are explicitly adhered to for human learning and instruction, an intermitent schedule of FB would be the most effective method. However, educational research had clearly shown that students correct more errors when FB is given for each error that occurs (Kulhavy, 1977).

Variables concerned with relief from physical need such as hunger, thirst, or sex were used to develop the explicit definition of the term, positive reinforcement. Students are not under the influence of physical deprivation in which FB fulfills a bodily requirement. FB does involve some affective elements related to positive reinforcement. For example, when a student is told they have made a correct choice, it is likely that they experience a degree of self-satisfaction. However, the primary function of FB is to provide information, understanding, and guidance (Gagne, Wagner, & Rojas, 1981).

FB helps learners to correct their errors by showing them where they went wrong and informing them of the correct action to take. Positive reinforcement will not be informative in cases where an error is made and more than two responses are possible. If there are several alternatives and an error is made, positive reinforcement (or the absence of it) will not point out what they should have done. FB is most valuable for revealing the correct action after an error has occurred (Kulhavy, Yekovich, & Dyer, 1976; Sassenrath & Yonge, 1968; Surber, 1975). In summary, FB plays a cognitive role in interaction between students and learning materials. FB is not synonymous with positive reinforcement and should not be associated with the same principles that guided early behavioral research.

Feedback Content

The type of information carried by FB effects how learners attend to and process information contained in a lesson. Kulhavy, Yekovich, and Dyer (1979) showed that if a review of text material was not part of the FB content, subjects would spend more time carefully reading the material prior to responding to MCs. The more concise FB that provides only knowledge of correct choice appears to have encouraged greater concentration on the instructional material.

Kulhavy et al. (1979) had subjects answer a number of MCs as they read the text material. Those subjects who received FB on the MCs made less errors than those that did not. Since none of the questions were repeated, practice was not a factor in the number of correct responses. This indicates that FB not only serves to correct errors, but improves learners' processing of information prior to assessment.

One possible explanation is that FB directs students' attention to key elements contained in the body of the text. Since FB's role is to provide information, it would seem that the more information it carried, the more efficient it would be. Surprisingly, this is not the case. Knowledge of correct response has been shown to be more efficient for error correction than FB that informs learners why their response was wrong (Gilman, 1969; Kulhavy, White, Topp, Chan, & Adams, 1985). It appears that learners attend mostly to what they perceive to be directly related to identifying the correct response. When additional content is added, such as text review, FB study time increases but without a significant improvement in error correction. Research is still inconclusive in explaining the effects of FB content. A greater understanding of this multifaceted variable is likely to require a thorough analysis of FB's influence on processing attention, level of comprehension, and the structure of information retained in memory.

Feedback Occasion

FB that immediately follows the learner's response has long been considered the most effective method of helping students correct errors. This staunch belief in immediate FB is strongly supported by another principle derived from research on reinforcement. Positive reinforcement increases the probability of a response reoccurring as a function of how closely it follows that response. FB's benefits are sometimes reduced when it is presented immediately. It has been demonstrated that FB delayed for 24 hours can be more effective for correcting errors than immediate FB (Kulhavy & Anderson, 1972; Sassenrath, 1975; Sassenrath, 1972; Surber & Anderson, 1975). This phenomena is called delay-retention effect (DRE) and has two contributing factors: (1) proactive interference, and (2) time spent studying FB.

When learners make an error on an MC followed by FB, their initial choice and the target can become confused in memory. This competition between one of the distractors and the target manifests as a repeated error on a later test when the learner must respond to the same question. When FB is immediate, their initial choice is still "fresh in memory" and likely to compete with the target on a later test. But, if FB is delayed, learners tend to forget the distractor, and this increases the probability of recalling the target. Kulhavy and Anderson labeled this phenomena as 'interference-perseveration' since errors tend to persevere over time.

The second contributing factor to DRE is defined in terms of effect of FB occasion on study behavior. Learners tend to spend more time studying FB after a delay. This is probably due to the decrease in frustration or tension after completing a challenging test or practice exercise. Surber and Anderson demonstrated that FB was more efficient in a normal classrooom setting if answers were given after a 24-hour delay rather than immediately following the test.

Learner Expectancy

What people believe that they will be held accountable for will effect how they perceive a task and how they perform on it (Bandura, Grusec, & Menlove, 1977). Sassenrath and Yonge (1968) found that informing subjects of a future test at the point of their receiving FB facilitated their retention of the material even when study time on the FB was held constant. Knowledge of increased accountability seems to have caused subjects to work harder at processing, or at least, memorizing the FB. This increased study effort produced significantly higher test scores.

A different type of learner expectancy was investigated by Kulhavy, et al. (1976). The amount of confidence subjects had when choosing an answer was assessed by their response to a five-point rating scale. A confidence rating was collected for each question. Subjects with high confidence ratings spent more time studying FB than lower confidence subjects when their answer was incorrect. High-confidence subjects spent the least amount of time studying the FB when their answers were correct. Compared to high-confidence subjects, low confidence subjects spent less time studying FB when an answer was incorrect and more time when their answer was correct. Overall, a high degree of confidence contributed to greater effort in utilizing the FB and a higher rate of error correction.

Conclusion

Considering the four variables outlined above, some recommendations can be made for FB in instruction and further research using MCs.

Frequency: FB should occur after every response to a question.

Content: FB should generally by as concise and brief as possible. Further research is necessary to derive methods for assessing the type of FB needed at the time that each error occurs.

Occasion: A delay in FB of approximately 24 hours should occur after a difficult test or practice session. More research is necessary to determine if the benefits of delayed FB are specific to the instructional setting or subject being taught.

Expectancy 1: Students should be informed that they will be tested over instructional material prior to study and practice activities.

Expectancy 2: Make sure that students have the entry skills for each lesson and are reasonably confident in dealing with the material prior to practice sessions.

Some FB problems and recommendations may by more easily addressed through the applications of CAI systems. For example, Kulhavy, et al. (1985) have indicated that simple response time may be a fairly good predictor of learner confidence. Adaptive CAI has been used to improve the learning of concepts by identifying and classifying types of errors during practice sessions (Tennyson & Park, 1980). Park and Tennyson (1986) point out that response analysis and moment-by-moment changes can be measured and utilized best with a computer. The consistent use of MCs in educational software should compel designers of CAI to apply well defined principles of FB. A discussion of computerized response timing, response analysis, and adaptive instruction are beyond the scope of this paper. However, the recommendations above may serve as a partial guideline for designing FB systems for CAI.

References

Bandura, A., Grusec, J. E., & Menlove, F. L. (1981). Observational learning as a function of symbolization and incentive set. In G. H. Bower & E. R. Hilgard (eds.), Theories of Learning (pp. 205-206). Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Gagne, R. M., Wagner, W., & Rojas, A. (1981). Planning and authoring computer-assisted instruction lessons. Educational Technology, 21(9), 17-26.

Gilman, D. A. (1969). Comparison of several feedback methods for correcting errors by computer-assisted instruction. Journal of Educational Psychology, 60(6), 503-508.

Kearsley, G., Hunter, B., & Seidel, J. (1983). Two decades of computer-based instruction projects: What have we learned. Technological Horizons in Education Journal, 10(3), 88-96.

Kulhavy, R. W. (1977). Feedback in written instruction. Review of Educational Research, 47(1), 211-232.

Kulhavy, R. W., White, M. T., Topp, B. W., Chan, A. L., & Adams, J. (1985). Feedback complexity and corrective efficiency. Contemporary Educational Psychology, 10(3), 285-291.

Kulhavy, R. W. & Anderson, R. C. (1972). Delay-retention effect with multiple-choice tests. Journal of Educational Psychology, 63(5), 505-512.

Kulhavy, R. W., Yekovich, F. R., & Dyer, J. W. (1976). Feedback and response confidence. Journal of Educational Psychology, 68(5), 522-528.

Park, O. & Tennyson, R. D. (1986). Computer-based response-sensitive design strategies for selecting presentation form and sequence of examples in learning of coordinate concepts. Journal of Educational Psychology, 78(2), 153-158.

Sassenrath, J. M. & Yonge, G. D. (1968). Delayed information feedback, feedback cues, retention set, and delayed retention. Journal of Educational Psychology, 59(2), 69-73.

Skinner, B. F. (1981). The behavior of organisms: An experimental analysis. In G. H. Bower & E. R. Hilgard (eds.), Theories of Learning (pp. 179-183). Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Surber, J. R. & Anderson, R. C. (1975). Delay-retention effect in natural classroom settings. Journal of Educational Psychology, 67(2), 170-173.

Tennyson, R. D. & Park, O. (1980). The teaching of concepts: A review of instructional design research literature. Review of Educational Research, 50(1), 55-70.

Thorndike, E. L. (1981). A proof of the law of effect. In G. H. Bower & E. R. Hilgard (eds.), Theories of Learning (pp. 135-136). Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Joseph Matoon
823 W. Brown St. #D
Tempe, AZ 85281
(602) 965-3359

SHUFFLE

An Intelligent System for Teaching Procedural Knowledge

Dr. Walter Maner

Bowling Green State University

ABSTRACT

This paper (a) presents the design and (b) explains the use of an intelligent and generative CAI system intended to teach and measure procedural comprehension. In this paper, SHUFFLE is shown to teach computer algorithms, but the domain of application is not limited to computer science and easily extends to any form of knowledge which can be represented as a series of steps and sub-steps, either done in sequence or in parallel.

SHUFFLE takes stepwise representations of procedural knowledge as input, randomly shuffles the steps to generate a scrambled problem of prescribed deviation, presents this problem interactively to the student, whose task is to unscramble the problem through a series of menu-driven edits. The system monitors the solution strategy pursued and, in that context, dispenses hints, diagnostics and next steps. At any point, the program's built-in "expert" knows the cut-and-paste operation which will result in the greatest progress, and can animate any part of a solution on the screen by running through a sequence of cuts and pastes. SHUFFLE copes effectively with most forms of indeterminism (parallelism) by recognizing multiple correct solutions.

Classroom use of pencil-and-paper versions of the scrambled problem test is compared to the interactive use of SHUFFLE, which models critical aspects of the manual method while overcoming several of its limitations.

INTRODUCTION

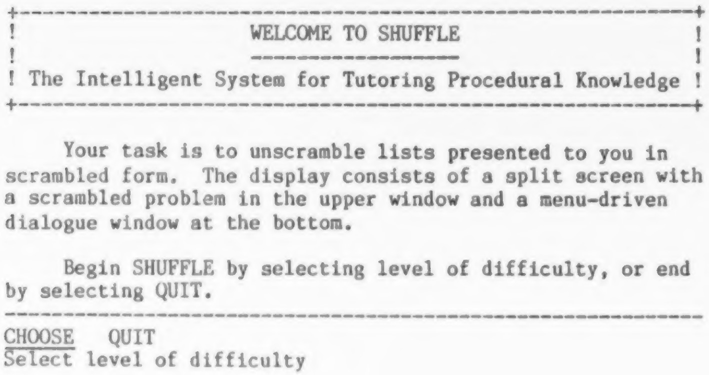
Although by no means restricted to the domain of computer science, SHUFFLE was originally intended to teach the specific form of non-declarative knowledge expressed in computer algorithms. Defining *algorithm* as "an effective *procedure*" clearly places procedural knowledge at the heart of the computer science curriculum. This creates the expectation that such a curriculum ought to make students better at algorithm analysis and design, among other things, which in turn should be reflected at practical levels in increased programming competence. The difficulty has been that we had no clear idea how to measure the partial correctness of the procedural knowledge students were expected to acquire, with the effect that we have been unable to certify programming competence (Sachman, 1968).[1]

The concepts and methods of this paper do at least mitigate this difficulty, not only for computer science but also for various other disciplines which aspire to transmit procedural knowledge. Toward that end, SHUFFLE can be regarded as delivering a new "question type" for use by existing script-driven instructional delivery systems -- systems which have heretofore been limited mainly to declarative query-and-response modes. SHUFFLE is, in fact, a component of the KNOWSYS[tm] instructional delivery system currently under development by the author.

HOW SHUFFLE EVOLVED: COMPARISON WITH MANUAL METHODS

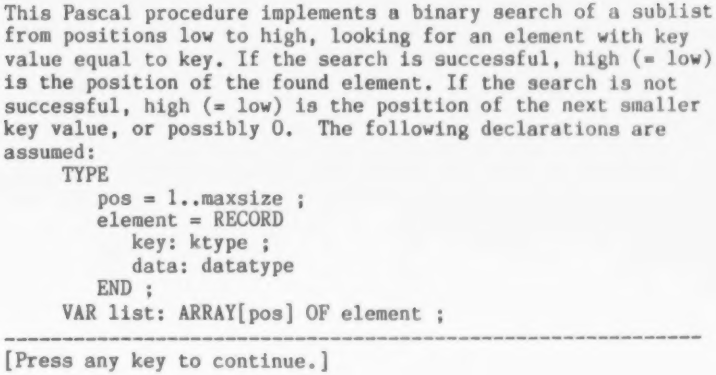
The author has been using scrambled algorithms as a vehicle for testing procedural comprehension since 1981. We have already learned from naive studies that success in unscrambling these problems correlates strongly with success in computer science generally,[2] but

Figure 1
Selection Screen for Choosing Level of Difficulty



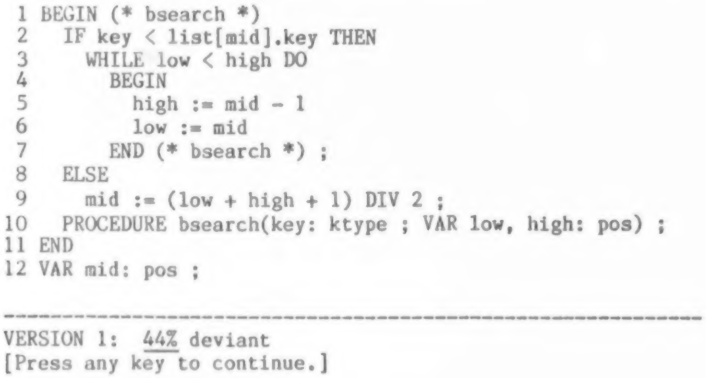
```
+----------------------------------------------------------+
!                    WELCOME TO SHUFFLE                    !
!                    ------------------                    !
! The Intelligent System for Tutoring Procedural Knowledge !
+----------------------------------------------------------+

     Your task is to unscramble lists presented to you in
scrambled form.  The display consists of a split screen with
a scrambled problem in the upper window and a menu-driven
dialogue window at the bottom.

     Begin SHUFFLE by selecting level of difficulty, or end
by selecting QUIT.
------------------------------------------------------------
CHOOSE   QUIT
Select level of difficulty
```

Figure 2
Problem Description Screen



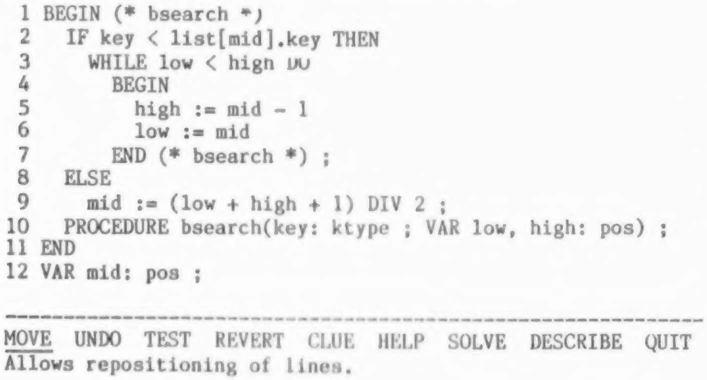
```
This Pascal procedure implements a binary search of a sublist
from positions low to high, looking for an element with key
value equal to key. If the search is successful, high (= low)
is the position of the found element. If the search is not
successful, high (= low) is the position of the next smaller
key value, or possibly 0.  The following declarations are
assumed:
     TYPE
        pos = 1..maxsize ;
        element = RECORD
           key: ktype ;
           data: datatype
        END ;
     VAR list: ARRAY[pos] OF element ;
------------------------------------------------------------
[Press any key to continue.]
```

Figure 3
Initial Problem Screen



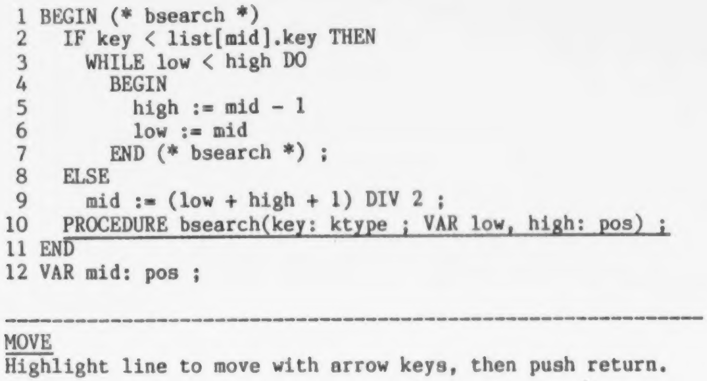
```
 1 BEGIN (* bsearch *)
 2   IF key < list[mid].key THEN
 3     WHILE low < high DO
 4       BEGIN
 5         high := mid - 1
 6         low := mid
 7       END (* bsearch *) ;
 8   ELSE
 9     mid := (low + high + 1) DIV 2 ;
10   PROCEDURE bsearch(key: ktype ; VAR low, high: pos) ;
11 END
12 VAR mid: pos ;

------------------------------------------------------------
VERSION 1:  44% deviant
[Press any key to continue.]
```

Figure 4
SHUFFLE's Main Menu Screen



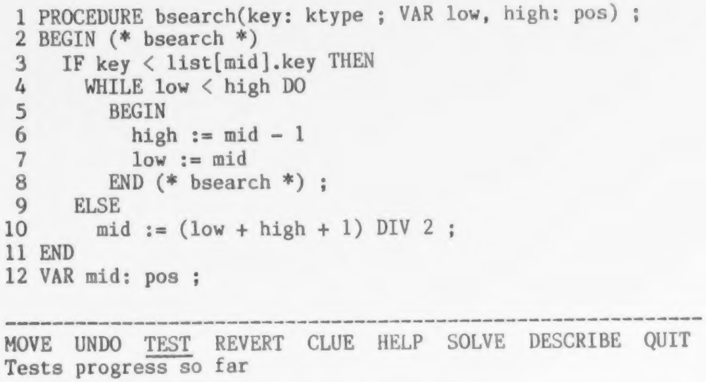
```
 1 BEGIN (* bsearch *)
 2   IF key < list[mid].key THEN
 3     WHILE low < high DO
 4       BEGIN
 5         high := mid - 1
 6         low := mid
 7       END (* bsearch *) ;
 8   ELSE
 9     mid := (low + high + 1) DIV 2 ;
10   PROCEDURE bsearch(key: ktype ; VAR low, high: pos) ;
11 END
12 VAR mid: pos ;

------------------------------------------------------------
MOVE  UNDO  TEST  REVERT  CLUE  HELP  SOLVE  DESCRIBE  QUIT
Allows repositioning of lines.
```

Figure 5
Edit Screen for Moving Lines



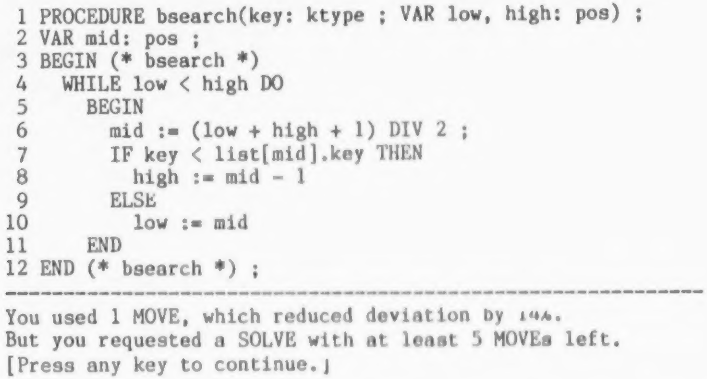
```
BEGIN (* bsearch *)
  IF key < list[mid].key THEN
    WHILE low < high DO
      BEGIN
        high := mid - 1
        low := mid
      END (* bsearch *) ;
  ELSE
    mid := (low + high + 1) DIV 2 ;
  PROCEDURE bsearch(key: ktype ; VAR low, high: pos) ;
END
VAR mid: pos ;
-------------------------------------------------------------
MOVE
Highlight line to move with arrow keys, then push return.
```

Figure 6
Main Menu Screen after Move



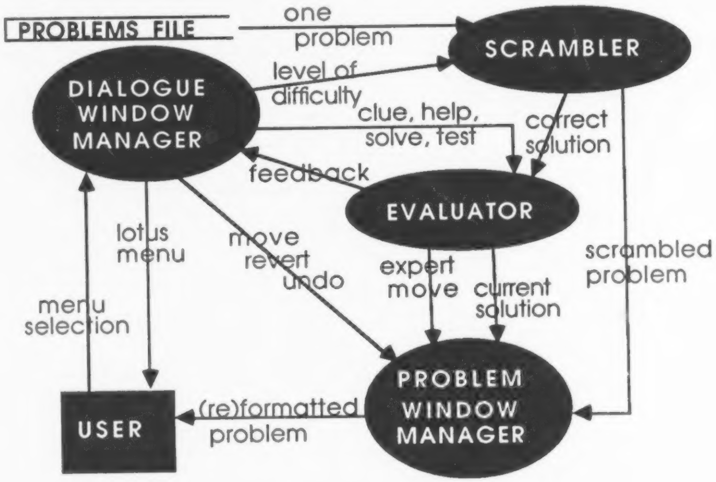
```
PROCEDURE bsearch(key: ktype ; VAR low, high: pos) ;
BEGIN (* bsearch *)
  IF key < list[mid].key THEN
    WHILE low < high DO
      BEGIN
        high := mid - 1
        low := mid
      END (* bsearch *) ;
  ELSE
    mid := (low + high + 1) DIV 2 ;
END
VAR mid: pos ;
-------------------------------------------------------------
MOVE  UNDO  TEST  REVERT  CLUE  HELP  SOLVE  DESCRIBE  QUIT
Tests progress so far
```

Figure 7
Final Diagnostics after On-screen Animated SOLVE



```
PROCEDURE bsearch(key: ktype ; VAR low, high: pos) ;
VAR mid: pos ;
BEGIN (* bsearch *)
  WHILE low < high DO
    BEGIN
      mid := (low + high + 1) DIV 2 ;
      IF key < list[mid].key THEN
        high := mid - 1
      ELSE
        low := mid
    END
END (* bsearch *) ;
-------------------------------------------------------------
You used 1 MOVE, which reduced deviation by 14%.
But you requested a SOLVE with at least 5 MOVEs left.
[Press any key to continue.]
```

Figure 8
Top-level Data-flow in SHUFFLE


PROBLEMS FILE
one problem
SCRAMBLER
level of difficulty
DIALOGUE WINDOW MANAGER
clue, help, solve, test
correct solution
feedback
EVALUATOR
lotus menu
move revert undo
expert move
current solution
scrambled problem
menu selection
PROBLEM WINDOW MANAGER
USER
(re)formatted problem

further controlled (longitudinal) studies are needed to determine whether continued practice with scrambled problems translates directly into continually improving comprehension. Research in this area has been held back because the pencil-and-paper versions of the task we wish to give to students are poor instruments indeed. For one thing, most students need to have each discrete step of the problem written on a separate strip of paper so that the strips can be physically rearranged, horizontally and vertically. And, once the student is satisfied with the rearrangement, he needs a foolproof system for converting what he sees on the desktop to a series of responses on an standard answer sheet. We have found both of these activities to be highly error-prone in their manual versions. Worse still, even with answer sheets in hand we will still have no idea how many moves were required to solve the problem, which is surely one of the most important indicators of competence. In addition, reverting to some earlier stage of the solution can't be done accurately and, in any event, the student has only one answer sheet, which means one chance to solve the problem. Finally, when we try to judge the correctness of a manual solution, it is easy to see when mistakes have been made, but it is often impossible to gather any precise sense of how wrong these specific mistakes cause the problem to be.

SHUFFLE resolves or mitigates nearly all of these difficulties. The student works at an organized "desktop" which gives him full control over the reassembly process. He cuts and pastes at will, with the system adding structure-revealing whitespace as needed. Reversion to a previous stage of reassembly is always correctly managed. Multiple try-and-test cycles are tolerated. And, best of all, we can always discover exactly how many moves were required to bring the problem into its final, possibly correct form, and we know how well these moves compare to those which would have been optimal at the time the move was executed. Given these measures and facilities, we should now be able to research procedural comprehension in a way that will have a major impact on the design of instructional delivery systems.

HOW SHUFFLE LOOKS: THE USER INTERFACE

As explained below, SHUFFLE takes stepwise representations of procedural knowledge as input, then randomly shuffles the steps to generate a scrambled problem of the requested deviation (difficulty).

[See Figure 1]

SHUFFLE presents this problem interactively to the student, beginning with the author-supplied description given in the problem file,

[See Figure 2]

followed by the problem itself. SHUFFLE is able to infer indent and undent rules from the solution given in the problem file by the author, and applies these rules to produce similar structure-revealing indentation in the scrambled version:

[See Figure 3]

The student has the task of unscrambling the problem through a series of menu-driven commands which allow lines to be repositioned. Choosing **MOVE** from the main menu

[See Figure 4]

activates sub-menus (some not shown) which allow a highlight bar to be positioned, first over the line to be moved, then over the insertion point:

[See Figure 5]

This produces a new display in the upper window, correctly re-indented to reveal the latest changes in substructure:

[See Figure 6]

The **TEST** option, which is about to be selected in Figure 6 (above), will inform the student that the problem is now 68% correct and that it can be solved in a minimum of five additional moves.

Other commands shown in the main menu (see Figure 6) have the following effects. **UNDO** undoes the most recent MOVE; a series of UNDOs will eventually put the problem back in its original form. **REVERT** returns the problem to the state it was in at the time of the last TEST and prunes the UNDO chain to agree. **CLUE** places a highlight bar over the most deviant line in the problem (e.g., line 12 in Figure 6). **HELP** causes the built-in "expert" to make the best MOVE possible. **SOLVE** is like a series of HELPs, which in turn generate a series of MOVEs, causing a complete solution to be animated on the screen. CLUE, HELP and SOLVE work by bringing intelligent heuristics to bear on abstract representations of the problem space. **DESCRIBE** allows the student to review the original description of the problem.

For any of these choices, typing the first letter of a menu option is sufficient to select that option. Alternatively, the left and right arrow keys can be used to position the menu strip highlight over the desired option; the return key will then select the highlighted option.

After a SOLVE, QUIT or user-produced solution, the following screen is displayed:

[See Figure 7]

At this point, the student can work through the same or a different problem, scrambled to the same or a different level of difficulty.

The behavior of SHUFFLE is summarized in the following data-flow diagram, where circles represent the major process components of the system and arcs represent the major data paths:

[See Figure 8]

HOW SHUFFLE WORKS: KNOWLEDGE REPRESENTATION ISSUES

Let us assume for the moment that procedures consist of discrete steps arranged in an uncomplicated linear sequence and that the construction or reconstruction of this sequence is good evidence of procedural comprehension. Then to test comprehension, we could require a simple *listing* of steps and count as fully correct those responses which included all and only those steps required, and in the correct linear order. But if, as is often the case, we need to measure *partial* correctness, then the missing, extra, merged or split steps which arise in open listing will make that measure very difficult to formulate. Some of the problems can be avoided by using *ranking* questions, which present all and only the steps required, but not in the correct linear order. While this is a help, it does not dispose of all the complicating factors. This is because it is common for procedural knowledge to exist in the form of *nested subsequences* and even in the form of *undetermined sequences*.

Subsequencing. To cope with nested sequences, SHUFFLE requires the author to supply an *outline* representation of the procedural knowledge which is to be taught, with subsequences indented relative to sequences. The steps belonging to each subsequence are enumerated separately. The author-supplied form of the "bsearch" example illustrates the technique:

Figure 9
Author-supplied From of "Bsearch" Problem

```
 1 PROCEDURE bsearch
     (key: ktype ; VAR low, high: pos) ;
 2 VAR mid: pos ;
 3 BEGIN (* bsearch *)
   1 WHILE low < high DO
   2 BEGIN
     1 mid := (low + high + 1) DIV 2 ;
     2 IF key < list[mid].key THEN
     3 high := mid - 1
     4 ELSE
     5 low := mid
   3 END
 4 END (* bsearch *) ;
```

Pronouncing a given reconstruction of an outline "partially correct" means than some vertical misplacement or some horizontal misplacement remains, or both. That is, some of the elements are in relatively correct order but at the wrong level, at relatively correct levels but in the wrong order, or both. A proper measure of deviation (or partial correctness) should take both vertical and horizontal displacement into account. SHUFFLE distinguishes these well enough to provide correctly indented displays of partially solved problems, but the scoring algorithm used to measure deviation at present collapses all displacement to vertical displacement:

```
FOR each line in the problem, say L, DO
BEGIN
  FOR every line above L which should be
      below L, add one to the deviation
      count
  FOR every line below L which should be
      above L, add one to the deviation
      count
END ;
```

Scoring algorithms which introduce *weights* into the calculation (for example, by multiplying the vertical displacement by the horizontal displacement) can be readily imagined and programmed, but in four years of experience we have not found one that gives results which square with students' intuition.

Indeterminism. In any given outline, there may occur a group of elements which have a place in sequence *as a group* but are unordered relative to each other. For example, in an outline containing the non interacting entries **A** to G, where the only constraints of order are that **A**, B and C must precede the partition **D** while **E**, F and G must follow it, we see that the sets **{A,B,C}** and **{E,F,G}** are undetermined in this sense. If **A**, B and C are understood to point to procedure steps, then indeterminism would permit the three steps to be executed in any order (or in parallel). Procedures in general, and computer algorithms in particular, contain a surprising amount of indeterminism, including nested indeterminacy, so a proper measure of deviation must allow for it. Accordingly, SHUFFLE requires that the author-supplied from of the problem identify non-deterministic steps:

Figure 10
Non-deterministic Author-supplied Problem

```
1  PROCEDURE quicksort(l, r: integer) ;
2  VAR v, t, i, j: integer ;
3  BEGIN (* quicksort *)
   1  IF r > l THEN
   2  BEGIN
      1  v:= a[r] ;
      1  i:= l - l ;
      1  j:= r ;
      2  REPEAT
         1  REPEAT i:= i + l
              UNTIL a[i] >= v ;
         1  REPEAT j:= j - l
              UNTIL a[j] <= v ;
         2  t:= a[i] ;
         3  a[i]:= a[j] ;
         4  a[j]:= t
      3  UNTIL j <= i ;
      4  a[j]:= a[i] ;
      5  a[i]:= a[r] ;
      6  a[r]:= t ;
      7  quicksort(l, i - l) ;
      8  quicksort(i + l, r)
   3 END
4 END (* quicksort *) ;
```

Indeterminism is by no means confined to computer algorithms. It is in fact more common in other forms of procedural knowledge, even recipes:

Figure 11
Non-determinism in a Recipe

```
1  Make cake
   1  Oil and flour cake pans
   1  Separate eggs
   1  Chop nuts
   1  Combine buttermilk and soda
        and let stand
   1  Mix sugar, butter and oil
   2  Beat egg whites until stiff
   2  Continue mixing, adding
        egg yolks one at a time
   3  Add buttermilk,
        soda mixture and vanilla
   4  Fold in egg whites,
        coconut and nuts
   5  Put mixture in baking pans
   6  Bake at 325 for 30 minutes
1  Make frosting
   1  Soften cream cheese
        until room temperature
   1  Soften 1 stick of butter
   1  Sift 1 box of powdered sugar
        into mixing bowl
   2  Add softened cream cheese,
        butter and vanilla to bowl
   3  Mix well
2  Frost cool cake
```

SHUFFLE's scoring algorithm has been extended to recognize multiple correct solutions arising from most, but not all, forms of indeterminism. The more extreme cases are not handled. For example, if the procedure shown in Figure 10 had contained the line "Writeln('Sort in progress ...');", then that line could be placed without error almost anywhere at any level. We have no way to represent such multi-dimensional "wildcards" at present. If, however, the line is "wild" only in the one-dimensional sense that it can occur at any place within a level, then we could require the author to mark it with an asterisk instead of the usual arabic number. Even these one-dimensional wildcards introduce enormous complexities and inefficiencies into the scoring algorithm, so they are not currently implemented in SHUFFLE.

Relative Deviation. Beyond subsequences and undetermined sequences there is a third complication: relative deviation. We want to know exactly how deviant a misordering we have, that is, what percent of the way it lies down some line drawn between a correct ordering and the most deviant misordering possible. We can expect a "best case" to be given by the author, but what would constitute a "worst case"?

Fortunately, the scoring algorithm given above in pseudocode has an easily understood worst case: an exactly backward misordering. So, if the problem contains N steps, then the worst-case deviation will be **N * (N - 1)**. That is, for every step in the problem, the maximum number of steps are misplaced, namely, all the steps above the step in question as well as all the steps below it. This is precisely what intuition would lead us to expect, and that compatibility with intuition is a most convincing reason for keeping the scoring simple. Thus, at any given time, we can know our position on a line drawn between the correct solution and the worst case simply by taking a ratio of the current deviation to this worst case.

HOW SHUFFLE WILL EVOLVE: PLANNED EXTENSIONS

(a). At present, SHUFFLE merely recognizes multiple correct solutions arising from indeterminism. Future versions will explicitly tutor indeterminism by asking the student to highlight, for each level, sets of steps which could be performed in parallel. This is of major importance in computer science, given the rise of declarative programming methodologies and the advent of profoundly parallel machine architectures.

(b). At present, SHUFFLE's built-in expert models only one type of solution strategy, which we will call the worst-first strategy:

```
1 WHILE NOT solved DO
2 BEGIN
  1 Use heuristics to find the most
    seriously displaced step
  2 Move it to a position that will
    produce the greatest reduction in
    global deviation
3 END
```

If other strategies were modelled, SHUFFLE's diagnostic messages could be improved. Strategies worth modelling include the two-window strategy

```
1 WHILE steps remain in window A DO
2 BEGIN
  1 Pick any step
  2 Move it to a correct position relative
    to other steps already moved to window
    B
3 END ;
4 IF NOT solved THEN use the worst-first
  strategy on window B
```

and the classify strategy:

```
1 Move to window B all the steps that
  force indents and undents, leaving the
  remainder in window A
2 Use the worst-first strategy on window B
  to get the structure right
3 Use the two-window strategy on both
  windows to fill in the minor details
```

Obviously, the optimum (or recommended move) on one strategy would not necessarily be the optimum move on another strategy.

(c). At present, SHUFFLE does not handle the more subtle forms of indeterminism, so we are devising better ways to flag "wildcard" steps in the problem file and better algorithms for calculating deviation when wildcards are present.

(d). At present, SHUFFLE equates deviation to difficulty in a simple-minded way. New versions will collect statistics to see whether this equation is valid.

(e). Some MOVEs are better than others, inviting tutoring, but SHUFFLE does no significant amount of tutoring at present. Newer versions will incorporate dialogue rules (Burton, 1982) which, together with better student modelling, should allow the program to capitalize on its tutoring opportunities.

(f). The task of "chunking" a problem into separate (hence, movable) lines falls entirely to the author at present. Syntax templates could help with this process in certain contexts.

REFERENCES

1. Burton, R. R. and Brown, J. S., "An Investigation of Computer Coaching for Informal Learning Activities," in Sleeman, D. and Brown, J. S. (eds.), Intelligent Tutoring Systems, Academic Press, New York (1982).

2. Sachman, H. et al., "Exploratory Experimental Studies Comparing Online and Offline Programming Performance," Communications of the ACM, 11 (1), (Jan., 1968).

ENDNOTES

1. If by competence we mean to include (at least) the basic ability to write correct computer procedures, then one might expect wide initial differences in individual competence to be smoothed by passage through a program of study. Apparently, training does not produce the expected smoothing; on the contrary, the more competent programmers studied were found to be 5 to 16 to 28 times better at writing correct procedures than the less competent ones (Sachman, 1968). If our methods of teaching and testing procedural knowledge are deficient, then this is precisely the result we should expect.

2. The Pearson correlation of (a) percent scores on pencil-and-paper scrambled problems with (b) cumulative grade point averages in computer science courses is statistically significant and typically exceeds .7 for juniors and seniors.

The Effects of Textual, Graphic, and Animated Graphic Elaboration Strategies and Practice on Learning from Computer-Based Instruction

Lloyd P. Rieber

Texas A & M University

ABSTRACT

There were two purposes to this study: 1) examine the effects of animation and practice on factual and application learning in a CBI lesson; and 2) investigate the extent to which animation and practice promote near and far transfer of these learning outcomes. Two levels of Practice (Relevant Practice, Irrelevant Practice) were crossed with three levels of Graphic Type (Graphic, Animated Graphic, No Graphic) and two levels of Text Type (Text, No Text). The CBI content was an elementary physics lesson in Newtonian mechanics. A total of 192 intermediate elementary school students participated as subjects. No main effects were found for the Practice, Text, or Graphic factors. Among the within-subjects factors (Learning Outcome, Transfer), a main effect was found for transfer: students performed significantly better on far transfer questions than near transfer questions. A significant interaction was detected between Learning Outcome and Transfer: far learning was facilitated for factual learning, whereas no difference in transfer was detected for application learning.

INTRODUCTION

Instructional design has evolved into a complex science involving the manipulation of many instructional variables. Effective delivery of instruction has consistently been the focus of much attention in instructional design. The recent availability of microcomputers has provided designers with many instructional alternatives as well as the need to consider important instructional decisions. The potential of the computer as an effective instructional delivery system has been carefully reviewed and studied (see, for example, Edwards, Norton, Taylor, Weiss & Dusseldorp, 1975; Bangert-Drowns, Kulik, & Kulik, 1985, and Clark, 1985).

Computer technology affords designers with a myriad of useful design options. Features such as immediate and informative feedback, lesson branching, lesson management, and visuals have been discussed and reviewed in length. Designers can also effectively incorporate a wide range of teaching strategies and activities into computer-based instruction (CBI).

Potentially unique features of CBI have been discussed and debated. Some, such as Clark (1983), have cautioned against concentrating too heavily on the features of any one particular medium. Clark has asserted that virtually all significant results found in CBI research stem from purely instructional variables and are only indirectly associated with the computer. Although controversy remains (see, Petkovich & Tennyson 1984 vs. Clark, 1983), it appears clear that many features of the computer deserve greater attention among researchers. Animation is one such feature which can be more easily incorporated into CBI than perhaps other delivery systems. Although instruction typically uses animation as an extrinsic motivator, there is a need to investigate the use of animation to directly or indirectly support or enhance instruction. An example of computer animation used directly in instruction is the manipulation of an animated block accelerating down an inclined plane to teach a concept in a physics class. Recent research on the effectiveness of animation as an aid to instruction has failed to demonstrate that animation improves learning (see, for example, Rieber & Hannafin, 1986; A.

Caraballo, 1985; J. Caraballo, 1985).

The lack of empirical support for animation presents an apparent discrepancy between hypothesized animation effects (as supported by a theoretical perspective of the importance of providing information in verbal and visual representations in promoting useful information encodings) and the empirical evidence to date. Research in the area of cognitive psychology points to the apparent retrieval advantages of multiple encoding at the time of learning. Mental imagery researchers (Bower, 1972 and Paivio, 1979) have recognized mental imagery as one of the primary mental structures, along with verbal representations. Many studies (Bower, 1972; Paivio, 1979; Dwyer, 1978) point to the power of pictures over abstract words for purposes of higher levels of retention. The problem seems to rest in the need to define appropriate contexts where the potentially powerful visual effect of animation can be used most effectively.

There were two primary purposes to this study: 1) examine the effects of animation and practice on factual and application learning in a CBI lesson; and 2) investigate the extent to which animation and practice promoted near and far transfer of these learning outcomes. It was hypothesized that students provided with verbal and graphic elaborations of lesson content would achieve higher posttest scores than groups receiving only verbal or only graphic elaborations. It was further hypothesized that practice would be of greatest value where the elaboration support was minimal.

METHODS

Subjects

The subjects consisted of 192 fourth, fifth, and sixth graders from a rural public elementary school. Participation was voluntary and selection based on parent consent. The subjects represented a typical cross-section of elementary school students. In addition, three fifth grade classes from another public school served as pilot subjects to validate the materials as well establish reliability on the testing instruments.

CBI Lesson Content

The CBI lesson used as the instructional content for this study was first developed for a previous study (see Rieber & Hannafin, 1986). This lesson described and explained Isaac Newton's Laws of Motion. The lesson material was divided into four parts: 1) introductory material; 2) motion resulting from equal forces in opposite directions in one dimensional space; 3) motion resulting from unequal forces in opposite directions in one dimensional space; and 4) motion resulting from equal and unequal forces in two dimensional space.

The first lesson part introduced students to Isaac Newton's formal discovery of certain physical laws. This part also initiated the sequence of learning activities used for the rest of the lesson. Most of the information presented in the first part was factual in nature. The second part introduced the concept that equal but opposite forces are needed to cause objects to stop. This section dealt only with one-dimensional space. The third part expanded this notion to include the effects of unequal forces acting upon a stationary object. The final direction and speed of an object is a combination of all of the forces from both directions. The third part also presented these concepts in one-dimensional space. Finally, the fourth part added the notion of two-dimensional space to the above concepts. Again, the final direction and speed of the object results systematically from the sequence of forces which act upon it.

All instruction was presented at an introductory level with the technical descriptions of the forces of gravity and friction removed. Approximately 45 minutes was required to complete the lesson.

Lesson Versions

Elaboration strategies were embedded at various locations within the lesson. The locations of these elaboration strategies were determined by the performance of students in a pilot study. The elaboration strategies were added to the lesson in areas found to be the most difficult and needing additional instruction. Two factors (Text Type, Graphic Type) were used to determine the nature of the embedded elaboration strategies: two levels of Text (Text, No Text) were crossed with three levels of Graphic (Graphic, Animated Graphic, No Graphic). The combination of these two factors gave the following treatment permutations: text only, text plus graphic, text plus animated graphic, graphic only, animated graphic only, no text/no graphic.

The text and graphic factors were crossed by a practice factor with two levels (Relevant Practice, Irrelevant Practice). Relevant practice consisted of four multiple-choice questions after each of the four lesson parts: two factual and two application. Feedback in the form of knowledge of correct results was provided. Irrelevant practice consisted of a noninstructional activity which required approximately the same time to complete as the relevant practice. The format of the noninstructional activity was also multiple-choice questions, however, the questions concerned information which was irrelevant to the lesson content.

Lesson Posttest

A 37-item posttest was developed which consisted of two types of questions: 16 factual and 21 application. These learning outcomes correspond to verbal information and rule-using, respectively, as defined by Gagne' (1977). A multiple-choice question format (1 answer and 4 distractors) was used to test the above learning outcomes. All test questions were presented to the students by computer.

A total of 19 questions measured near transfer and 18 questions measured far transfer: 9 factual questions and 10 application questions measured near transfer; 7 factual questions and 11 application questions measured far transfer. Overall test reliability, as determined during the pilot study, was .91. An example of near transfer would be a question presented in the context of a ball being kicked. An example of far transfer would be a question presented using a wooden block instead of a ball.

Procedures

In order to determine the content validity and reliability of the materials as well as to substantiate the instructional design of the materials, a pilot study was conducted using a small sample of fifth grade students. In this pilot study, the four instructional parts (introductory, equal forces in one dimension, unequal forces in one dimension, equal and unequal forces in two dimensions) were presented in an all-text form without any embedded elaboration strategies. Data from this pilot study was used to determine the validity and reliability of the test items in order to help determine the final selection of appropriate test items. Also, this data was used to determine where supplemental instruction in the form of the elaboration strategies was most needed by students.

In the main experiment, students were randomly assigned to one of the 12 treatment conditions. All treatments and data were delivered and recorded by computer. On average, approximately 45 minutes was needed to complete the lesson.

Design and Data Analysis

In this study a 3 X 2 X 2 X (2 X 2) completely crossed factorial design was used. Three between-subjects factors (Graphic Type, Text Type, and Practice) were crossed with two within-subjects factors (Learning Outcome, Transfer). The between-subjects factors consisted of: Graphic Type (Graphic, Animated Graphic, No Graphic); Text Type (Text, No Text); and Practice (Relevant Practice, Irrelevant Practice). The within-subjects factors consisted of: Learning Outcome (Fact, Application); and Transfer (Near, Far). A mixed factor analysis of variance (ANOVA) was chosen as an appropriate statistical procedure for the study. This analysis is commonly known as a repeated measures analysis of variance (Winer, 1971). The two within-subjects factors also served as the dependent measures.

RESULTS

No main effects were found for Graphic Type, $F(2,180)=.393$, $p=.676$, Text Type, $F(1,180)=.072$, $p=.789$, or Practice, $F(1,180)=.234$, $p=.629$. No main effect was found for Learning Outcome, $F(1,180)=2.285$, $p=.132$. A main effect was found for Transfer $F(1,180)=4.439$, $p=.037$: learners performed significantly better on far transfer questions than near transfer questions.

No interactions were detected among the between-subjects factors (Graphic Type, Text Type, Practice). However a significant interaction was found for the within-subjects factors (Learning Outcome, Transfer), $F(1,180)=3.911$, $p=.05$. The lesson promoted far learning for factual information, whereas far learning was not facilitated for the acquisition of application learning. A moderately significant interaction was found between the factors of practice and learning outcome, $F(1,180)=3.394$, $p=.067$. When provided with practice, students performed significantly higher on application questions than factual questions.

DISCUSSION

The lack of main effects among the embedded elaboration conditions was surprising. Previous research has demonstrated the power of pictures over abstract words in facilitating understanding of abstract concepts. Some mental imagery researchers have posed that information is encoded into long term memory in two ways: as a mental image and as a verbal representation (Paivio, 1979).

It was believed that learners would recall greater amounts of factual information as well as be more successful in solving application questions when provided with instruction which contained both graphic and textual elaboration cues than when no such cues were provided or when these cues provided only graphic or only textual information. The results from this study do not support this hypothesis.

There are several possible explanations for the outcomes found in this study. First, it is possible that the learners did not attend to the elaboration strategies. That is, although the elaboration strategies provided the potential for additional cognitive processing of the lesson information, the learners did not make use of it. Reasons for this are debatable. One probable explanation concerns the nature of the lesson content. The lesson material may have proved too difficult and complex for the students. This level of complexity could have prevented learners from mentally processing the elaboration strategies in the ways in which they were intended. A second explanation is that the embedded elaborations were instructionally too weak to affect learning. Since the elaboration strategies were not included in the validation process, it is difficult to make inferences about their effects.

The significant difference between near and far transfer learning is somewhat surprising. Typically, learning in near transfer situations would be expected to be greater than far transfer situations. One explanation is that the concepts were encoded meaningfully into "real life" contexts. These contexts were natural extensions of the "kicked ball" analogy used in the lesson. This rationale also helps to explain the significant interaction between learning outcome (Fact, Application) and transfer. Learners performed significantly better on factual questions in far transfer settings than on application questions in near transfer settings. The implication is that learners were actively elaborating on the ball example in simple, factually related ways and never fully processed the information in order to use it in solving application questions. If the elaboration strategies had been effective, as hypothesized, more consistent findings would have been expected.

The lack of main effects for practice was also surprising. Again, this result is probably an artifact of the task difficulty. It tends to support the conjecture that the lesson material was too complex for the age group. The practice activities were intended to provide opportunities for students to process relevant lesson content. These activities should have helped to simplify the lesson material in order to help students select and process relevant content. Furthermore, the activities should have encouraged organization and integration of content with prior knowledge and other lesson content. Rather, they appeared to increase the cognitive strain imposed on the students. Students probably perceived the lesson as too demanding and may have chosen to ignore potential practice benefits. Another explanation, though less likely, suggests that insufficient practice was provided. Asking students only four questions after approximately 11 computer frames may have been insufficient to activate adequate processing. No general interpretations of the usefulness of practice as an instructional technique should be inferred. The effectiveness of adjunct questions as practice strategies, under certain guidelines, has been well documented (Hamaker, 1986; Salisbury, Richards, & Klein, 1985; Wager & Wager, 1985).

Although an interaction between practice and elaboration strategy was expected, no interaction was observed. It was expected that when learners were provided good elaborative cues to facilitate appropriate imagery and verbal encoding, they would not rely on the cognitive processes activated by routine practice. Conversely, in the absence of the lesson cues believed to facilitate appropriate levels of encoding, other techniques, such as practice, would be necessary to achieve similar encoding levels. The absence of this interaction is also probably attributable to previous explanations: either the students did not attend to the lesson information appropriately, or the lesson material was too difficult.

This study has raised several questions concerning the relationship between student thought processing, instructional delivery systems, and instructional strategies. Due to an apparent confounding caused by difficult lesson content compounded by relatively ineffective lesson activities, the question of how computer animation may be used to positively affect learning remains largely unanswered. Future researchers should be careful to adequately match the complexity of materials with the maturity of the subjects. Also, replications should systematically validate the effectiveness of additional lesson activities. Lessons should be designed to direct the learner to both attend to, and use, lesson activities appropriately.

REFERENCES

Bangert-Drowns, R. L., Kulik, J. A., & Kulik, C.-L. C. (1985). Effectiveness of computer-based education in secondary schools, Journal of Computer-Based Instruction, 12(3), 59-68.

Bower, G.H. (1972). A selective review of organizational factors in memory. In E. Tulving & W. Donaldson (eds.), Organization and memory. New York: Academic Press.

Caraballo, A. (1985). An experimental study to investigate the effects of computer animation on the understanding and retention of selected levels of learning outcomes. Unpublished doctoral dissertation, The Pennsylvania State University, University Park.

Caraballo, J. (1985). The effect of various visual display modes in computer-based instruction and language background upon achievement of selected educational objectives. Unpublished doctoral dissertation, The Pennsylvania State University, University Park.

Clark, R. E. (1983). Reconsidering research on learning from media. Review of Educational Research, 17(2), 92-101.

Clark, R. E. (1985). Confounding in educational computing research. Journal of Educational Computing Research, 1(2), 137-148.

Dwyer, F. M. (1978). Strategies for improving visual learning. State College, Pa.: Learning Services.

Edwards, J., Norton, S., Taylor, S., Weiss, M., & Dusseldorp, R. (1975). How effective is CAI? A review of the research. Educational Leadership, 33, 147-153.

Gagne', R. (1977). The conditions of learning (3rd ed.). New York: Holt, Rinehart and Winston.

Hamaker, C. (1986). The effects of adjunct questions on prose learning. Review of Educational Research, 56, 212-242.

Paivio, A. (1979). Imagery and verbal processes. Hillsdale, NJ: Lawrence Erlbaum Associates.

Petkovich, M.D. & Tennyson, R.D. (1984). Clark's "Learning from Media": a critique. Educational Communications and Technology Journal. 32(4), 233-241.

Rieber, L. & Hannafin, M. (1986). The effects of textual and animated orienting activities and practice on learning from computer-based instruction. Submitted for publication.

Salisbury, D., Richards, B., & Klein, J. (1985). Designing practice: A review of prescriptions and recommendations from instructional design theories. Journal of Instructional Design, 8(4), 9-19.

Wager, W., & Wager, S. (1985). Presenting questions, processing responses, and providing feedback in CAI. Journal of Instructional Development, 8(4), 2-8.

Winer, B. (1971). Statistical principles in experimental design (2nd ed.). New York: McGraw Hill.

Lloyd P. Rieber
302 Rackley Building
University Park, PA 16802
(814) 863-2427

ONLINE TRAINING FOR TRAINERS: A WORKING MODEL BASED IN UNIX*

John A. Scigliano, Barry A. Centini, & Jacques Levin, Nova University

Alan White, Integrated Communications Technology, Inc.

ABSTRACT

Distance learning, CBT, learning by doing, reactive learning, adult learning theory and UNIX. What do they have in common? These concepts and technologies have been successfully integrated into an accredited electronic learning network. The program's developers and a recent graduate discuss key dimensions in the design and implementation of the program.

INTRODUCTION

A paradox found in the education of trainers is that the value of computer-based learning is stressed by instructors working in traditional classroom settings. By learning through a computer-based online delivery system the student is able to study both the method of course delivery and the content, and becomes an active part of the learning process. Distance learning through computer networks has become more than just an option for some students, it is the only way they can pursue their academic career without total personal and work upheavals. Traditional residence programs can only meet the needs of students within commuting distance, creating a need for programs that can deliver training by non-traditional methods regardless of place of residence. And for trainers who will be developing computer-based learning systems, using online distance learning as the primary delivery system for their education reinforces this need.

The field of computer-based training is developing rapidly. Advances in the fields of digital computers, interactive video, CD/ROM, expert systems, and telecommunications make staying "current" a challenge for trainers. It is even more of a challenge for trainers in work settings or with family commitments that prevent them from attending resident programs. A possible alternative for these students is a field-based program designed specifically for the working professional.

The Doctor of Arts in Training and Learning (DATL) and the Master of Science in Training and Learning (MSTL) at Nova University are examples of such an alternative. They were developed by the Center for Computer-Based Learning and coexist with a similar program for information scientists (1,2). By providing learning environments that use the full power and flexibility of microcomputers, minicomputers, the UNIX operating system, and telecommunication networks these programs provide educational programs for students and allow them to work without the constraints of time and distance.

CURRICULUM

The curriculum in the online training programs reflect three aspects of cbt development: UNIX-based learning systems, artificial intelligence and expert system development, a concern for human factors. The following course list reflects this: (HF=Human Factors, AI= Artificial Intelligence, Expert Systems)

Courses in the D.A.T.L. Program

1. Digital Computers (UNIX, AI)
2. Research (HF)
3. Database Management *UNIX, AI)
4. Instructional Systems (UNIX, AI, HF)
5. Human Factors (HF)
6. Courseware Development (AI)

One of the requirements for admission to these programs is full-time employment in the training field. In keeping with this, the practicum process is designed to allow students to investigate a situation directly related to activities within their own institution or organization. The requirements are simple, the project must be a practical application of the course content in the student's work environment.

---

*UNIX is a Trademark of AT&T Bell Laboratories

A sampling of Practicum titles can give a flavor of the types of projects involved in this process:

- ° An Expert System Concept and Design to Train Parallel Processor Programmers
- ° Analysis of Potential Computer Systems to be Used To Deliver the Army's Basic Skills Curriculum with Recommendations for One System
- ° An Investigation into the use of an Authoring System for Curriculum Delivery using the UNIX System as a Network Delivery System
- ° Front-end Analysis and Instructional System Design for a Course on how to Perform as an OPNAV Action Officer
- ° Computer Graphics: A Guide to Selecting and Using a PC-Based Graphics System for Presentation Graphics in Training
- ° An Investigation of the Human Factors Aspects in a Telecommunication Network for Technology Transfer: A Feasibility Study

The Major Field Project (MFP) is the capstone of the three year program, and is also directly related to bringing about improvements or contributing research knowledge "at work". The following titles from the first three MFP's to be completed can give a sense of the types of projects undertaken:

- ° Prolog in Computer-Based Learning, A General Purpose Interface for Overcoming Learning Blocks
- ° An Expert System Prototype for Computer-Based Training
- ° Development of a Micro-Computer Based Knowledge System Software Package to Assist in the Preparation of Behavioral Learning Objectives

These programs stress the involvement of the student in learning "behind the scenes" tools and techniques used in delivering their education. The UNIX system has been criticized for being "too open" but in an environment like this that flaw is a definite strength.

SOFTWARE DEVELOPMENT

In the nearly four years of existence of the online programs of CBL, several areas of the learning environment provided enough challenges to warrant unique software development. By using UNIX shellscripts and C programs, the staff produced the following software tools:

1. ecr -- The "electronic classroom", an interactive learning environment that can accommodate up to 24 students per classroom. The design and application of this program have been discussed elsewhere [3], but the implications for trainers are unique in that students can use the facility as teacher or student and be ever aware of the design considerations from both settings.
2. electronic management tools -- A whole series of programs have been developed for filing student assignments, etc. and keeping track of progress. A system is under development that will allow for automatic storing and record keeping, with proper notification, to be in effect at all times with minimum supervision.
3. electronic curriculum guides -- Menus, tutorial drivers, tutorial expert systems, and other student aids have been designed to allow students to choose the level of "system help involvement" and find the electronic learning style that best fits them. A knowledge base has been eveloped for use by students and teachers. The knowledge base has both an inference engine driver to allow students/teachers to query it and also an expert system that automates the inclusion of new knowledge into the database. Again, it is the "behind the scenes" aspect of this program that allows the student to learn, and learn about their learnings, that has proven most beneficial.

STUDENT'S PERSPECTIVE

Perhaps the best way to describe the impact of the program through a student's (Alan White) eye is to include the review questions and imbedded evaluation from an assignment from the course on Courseware Development:

Review Questions:

1. Develop a one paragraph summary of your activities on this assignment indicating the key points that you have learned.

> ICAI, making significant use of recent advances in cognitive science and artificial intelligence, can provide qualitatively new type of learning environments. To realize this potential, however, investigators and instructional designers will require imagination along with a willingness to understand the social, cognitive, and motivational aspects underlying expertise, problem solving and learning.

2. What relationships can you identify between the work required in this assignment and other work that you have done in the program.

> This assignment brings together concepts learned in Instructional Systems, Human Factors, Research and Statistics, and Learning

Theory. It logically belongs first in the Courseware assignment sequence and toward the end of the Nova DATL sequence of studies.

3. What theories can you identify that relate to the work required in this assignment?

Most of the cognitive learning theories are applicable. Also, John Dewey's learning by doing concepts are directly applicable to discovery learning.

LESSONS LEARNED

The development, evaluation, and improvement of the DATL program presents an interesting challenge to educators experienced in traditional learning environments. Some lessons learned are:

1. Online learning does not work the same for all students. Measures of learning styles can be correlated with performance measures to predict success. The small samples used in early retention studies indicate complex interactions between styles and online work.
2. The UNIX environment must be modified regularly through systems programs to allow staff and faculty to adapt to changing student needs.
3. Faculty must have both traditional classroom skills and computer competencies.
4. Live interaction with the student is essential. The software tool "ecr" was developed expressly for this purpose and has proven very beneficial in providing a sense of "parallel learning" that counters any sense of isolation that may develop in the individual learner.
5. Development of expert systems to facilitate classroom management and provide prompt response to students is a necessity.

CONCLUSIONS

The Nova University doctorate and master's programs for trainers serves the needs of those students willing to pursue a graduate degree but unwilling to disrupt their family life. It offers an opportunity to upgrade skills in CBT and join a challenging educational experience without taking time away from their profession. Also, because it is a national program, the student who relocates does not undergo an interruption of schooling. The process of learning becomes the product, preparing the student for total involvement in the electronic learning communities of the future.

An obvious "side effect" of the development and implementation of this program is the inclusion of the trainer-as-student in these projects. Competencies gained from these programs cannot be measured by outcomes that only address course objectives; but we have yet to develop a method of evaluating the competencies and tools gained solely from participating in a continuously developing online CBT environment.

NOTES

1. J. Maddock and B. Centini, "Networked Training for Information Scientists". _Proceedings of the 49th Annual Meeting of the American Society for Information Science, Vol. 23:178-187; 1986_.
2. J. Scigliano and B. Centini, "UNIX in a Community of Scholars". Paper presented to the Mid Year Conference of the American Association of Information Science, Fort Lauderdale, Florida, May 1985.
3. J. Scigliano, D. Joslyn, and B. Centini, "A Real-Time UNIX-Based Electronic Classroom". Paper presented to the IEEE Southeastcon '87, Tampa, Florida, April 1987.

Barry A. Centini
Nova University
3301 College Avenue
Ft. Lauderdale, FL 33314
(305) 475-7047

Children Using Educational Software:

An Observational Study

Rosemary E. Sutton

Cleveland State University

Abstract

This paper is based on a qualitative observational study of children's use of commercial software. Examples of strategies that students developed while using educational software are described. These strategies for software use frequently required little thinking and subverted the intentions of the software designers or classroom teachers. The influence of the classroom organization and teachers' beliefs about computer use on these strategies are summarized. Finally, implications for software designers, researchers, and teachers are suggested.

Introduction

Thousands of educational software programs are on the market and are being used by children in the schools. Many studies have been conducted on the effects of computer-aided instruction on pupil's achievement and attitudes (e.g., Bangert-Downs, Kulik & Kulik, 1985; Clark, 1985; Kulik, Kulik & Bangert-Downs, 1985), but few researchers have studied children's behavior while using the software (some examples are Neuman, 1986; Sheingold, Hawkins, & Char, 1986, Trumbull, 1986).

This is an important omission for several reasons. First, we know that children make sense of their world by inventing their own rules and theories for understanding learning tasks and that frequently these rules and theories are not those intended by the teacher or curriculum designer (Easley & Zwoyer, 1975; Brown & Burton, 1984). This means that we can expect children to create their own rules when using software and that these rules may not be consistent with the objectives of the teacher or software designer (Turkle, 1984). Second, educational software is typically designed for use by one student but students use software in a social environment that frequently requires them to share computers, interact with other children and the teacher, learn under time constraints, work in a noisy environment, etc. (Sheingold, Hawkins & Char, 1986). This may influence how the software is used, and thus what learning is taking place.

Understanding how children use commercial software in a school environment cannot be gained by post-test achievement or attitude data, but requires observation of the children and the teachers in the classroom setting. The purpose of this paper is to report such an observational study conducted in two schools and to consider implications for teachers and software designers.

Schools A and B are part of a suburban, middle class, predominantly white school district in Ohio. The principal of school A tells all parents that every child in the school will use a computer during the course of the year. The school has eight Apple IIe's with color monitors in a "lab" which consists of half an empty classroom (the other half is assigned to tutoring learning disability students and to teaching gifted students), and six more Apple IIe's in individual classrooms. The school owns about 180 pieces of commercial educational software (including all the MECC[1] software). Every class above Grade 1 is assigned to the computer lab twice a week for approximately 25 minutes. During this time, half of the children go to the library, the other half go to the computer room with the classroom teacher. I spent one day a week for six weeks in the school, observing in the computer lab and in regular classrooms.

School B also has made a commitment to parents that the pupils will use a computer during the course of the year. The school has turned a classroom into an attractive computer lab containing 15 Apple IIe's with color monitors, and two printers. Parents helped raise money for the computers. The school owns lab packs of about 10 pieces of software, as well as multiple copies of selected MECC software. Classes are scheduled into the lab twice a week for about 30 minutes. I spent two half days observing in the computer lab.

Children's Strategies for Software Use

Two 6th grade girls were using Snooper Troops (Snyder, 1982). This problem solving game requires "driving" from place to place to search for clues in order to solve a crime. Solving the crime requires map drawing, keeping records, and takes several hours. Games can be saved at any time. When driving the user must "turn" (press a different key) at corners otherwise the car will crash and have to be sent to the garage for repairs. Initially, it is hard not to accidentally crash the car as the program has a slow response time. Repairs require nearly one minute of waiting before the game continues. About two minutes after beginning the program, the girls accidentally wrecked the car twice. Then, they spent more than five minutes deliberating crashing the car, waiting for repairs, then deliberating crashing the car again, waiting for repairs, etc. Obviously no problem solving was occurring.

The Incredible Laboratory (Kosel & Carlson, 1984) from Sunburst is a very popular problem solving program. Colorful monsters are created from different ingredients and the purpose of the program is to determine which part of the monster (e.g., head, body, foot) is made from which ingredient (e.g., snake oil). To do this, students choose which ingredients to use to create the monster and must systematically isolate one variable, observe the results, and keep records. I watched three pairs of children use this program and no child kept records. One pair spent 15 minutes making colorful monsters and making comments such as "Look at this! It has a green snake head". No problem solving was occurring.

The Pond (Kosel & Fish, 1984) is a problem solving piece of software designed to teach patterns. There are lily pads in a pond that a frog has to jump across to get to the other side. The user must determine the pattern (e.g., 4 left, 2 up, 1 down) that has to be repeated in order to get across the pond. One 4th grade pupil was observed creating a circular pattern (1 up, 1 left, 1 down, 1 right) and then watching while the frog jumped around and around in circles. Obviously some problem solving was occurring, but not what the designer had intended.

It can be that argued flexible problem solving software lends itself to pupils inventing their own (non-educational) strategies more than drills and that teacher supervision is more important in problem solving software. Advertisers of drills often claim that their software can be used with very little teacher intervention or supervision. But the best example of a pupil generating a strategy occurred while a 6th grade boy was using the multiplication drill called Dragon Mix (Chaffin, Maxwell & Thompson, 1985) To play the drill, the user must determine which of three pairs of numbers when multiplied give the number in a dragon's stomach, aim the mouth of the dragon at the correct pair, and then shoot. For any given number in the dragon's stomach, the product pairs get closer and closer to the dragon and if they get too close they will blast the city where the dragon lives. This means that speed is important. A 5th grade boy was using the program by himself. He had a very efficient strategy. He systematically aimed the mouth of the dragon at the top number pair and shot, then he aimed at the middle number pair and shot, then he aimed and shot at the bottom number pair, and then he aimed and shot at the middle number pair, then the top pair and so on. This strategy required no calculations so he did this very quickly and very rarely did any number pair get close enough to blast the city. At the end of a game, the boy looked for his score and his strategy was reinforced. The first number on the score screen was the number of tries or the number of times the user shot at the number pairs. Obviously, the number of times was increased by rapid, non-thinking firing. In subsequent lines on the screen, the number correct and percentage correct were given, but the boy paid little attention to these figures and only pointed the top score to his friend at the next computer. His strategy appeared to be excellent and he taught it to another boy. Obviously this boy was not practicing multiplication.

A Kindergarten boy using Train on Elementary Volume 7 (Kosel & Fish, 1981), an alphabet drill also developed his own non-thinking strategy. An example of a drill is the series letters e.g., **g h ... j k** placed on the screen and the child is rewarded for pressing the missing letter **i**. If the child does not get the answer correct on the first try the complete alphabet is presented on the bottom of the screen as an aid. Most of the class knew the alphabet much better than the keyboard layout so they had to hunt for the correct letter. One boy did not attempt to figure out the correct answer. Rather he just ran his hands over all the keys on the keyboard until the correct key was pressed. As there was no penalty in the program for guessing, he always got the correct answer.

Classroom Organization and Teacher Expectations

All the examples described above, except for the last one, were observed in school A. The organization of the computer lab, and teachers' expectations about computer use encouraged some of these non-thinking strategies. The computers in the lab were not joined by a network and there were no lab packs of software. This meant that in the lab there were typically 8 different programs being used simultaneously, which made effective teacher supervision very difficult. As well, when pupils had finished at the library, they drifted to the lab and sat on the floor reading books. The teachers frequently paid attention to these children. Because the computers in the lab were in half a classroom teachers were reluctant to do any formal introduction of a piece of software in case they interfered with the tutoring a few feet away.

The non-thinking strategies were also reinforced by the teachers' attitudes about computer use in this school. They perceived it as a non-planning time for themselves and a "fun" time for the children. They did not insist pupils bought paper to the computer lab even though many drills and problem solving programs require the use of pencil and paper. One teacher half way through the lab time typically moved all the pupils from one computer to another. She explained to me that pupils had a very short attention span on the computer so she moved them to a different piece of software. Pupils were moved even if they were involved in solving a problem. Another teacher tried to dissuade two 5th grade girls from working on a program, Code Quest (Hermann, Kosel & Sweeder, 1985), as she thought it was too hard. The girls really wanted to figure out the program (I had helped them get started) and they worked on one puzzle away from the computer during recess.

Some teachers did use the computer differently. For example, a first grade teacher planned a lesson by first demonstrating the program to the whole class and then assigning them to groups to work on it. The program was one that would load in multiple computers so all the children could work on the same problem simultaneously.

Computer use in School B was different for a number of reasons. First, the computers occupied a whole room. This meant all the students in a class could use the computers at once (children worked on the computers in pairs) and that the teachers could provide whole class instruction as they were not worried about the noise. Second, the school owned multiple copies of software so that typically only one to three programs were being used during one class period. This made teacher supervision much easier.

Teachers' attitudes towards the computer were also different. Computer use was not thought of as a "non planning" time. Several teachers told me about activities they did to prepare the children for a computer program. Students brought paper and pencils to class when it was necessary. Before a new piece of software was introduced the teachers typically demonstrated the program to the whole class to show the menu options and rules for operation.

In this second school, it appeared that less time was wasted with the pupils waiting for a teacher (or parent aid) to explain the program and with children having to wait for the program to re-boot because they accidentally exited the program. Also, fewer non-thinking strategies were observed. It may have been that I saw fewer non-thinking strategies because I spent less time observing in this school but I believe that the physical situation and teacher behavior influenced this.

Implications

Obviously more observational research is required before firm implications can be made but this study does suggest some implications for software designers, educational researchers and teachers. First, designers should field test their software in real classroom situations and observe children. Obviously, there is no way to predict all the non-thinking strategies children will use and that all software cannot be designed in such a way as to eliminate all possible misuses. However, the strategy used by the boy in Dragon Mix (Chaffin, Maxwell & Thompson, 1985) is relatively obvious and the strategy used by the Kindergarten boy with the alphabet drill could be avoided with a better design.

Second, designers and publishing companies should not advertise that their software can be used with no teacher supervision or intervention. Rather, written materials accompanying educational software should provide guidelines for use and possible problems to avoid. Some software programs seem to provide more potential for non-thinking strategies than others. For example, there appears to be little opportunity to misuse Factory (Kosel & Fish), a popular program which requires mental manipulations of three dimensional products. Many teacher reports confirm this impression. In contrast, my observations (both in School A and in other settings) suggest the potential for misuse in The Incredible Laboratory (Kosel & Carlson, 1984) is strong, and teacher reports are consistent with my observations.

Third, researchers, designers and teachers need to distinguish between the promise and the reality for the use of the computer in individualizing instruction. One of the greatest benefits cited for computer-aided-instruction in schools is the ability for each child to work at his or her own pace (Bork, 1987; Siegel & Davis, 1986). Research on non-computerized individualized instruction has indicated a number of practical problems. The most serious appears to be the demands on the teacher's time; the time required to foster student independence in staying on-task and the time to provide enough instructional supervision (Wang & Lindvall, 1984). Are the same problems evident when using CAI, or does the nature of the software minimize this problem? How much time is wasted while the child waits for the teacher to explain the use of the software and rules for operation, (e.g., how to escape from a program)? Do more children develop non-thinking strategies when using educational software than in a regular classroom? What instructional practices are most suited when using educational software? Obviously, further research is needed to determine this.

Fourth, teachers need to be aware of the messages they are giving their pupils about computer use. Mere exposure to "high quality" software without planning, instruction and an environment that encourages thinking will not guarantee that pupils are mindfully engaged. Classroom use of educational software does not diminish the importance of the role of the teacher but may well increase it.

Notes

[1] MECC (Minnesota Educational Computer Consortium) software can be purchased with a site licence that allows multiple copies to be made in each school building.

References

Bangert-Drowns, R. L., Kulik, J. A., & Kulik C-L, C. (1985). Effectiveness of computer-based education in secondary schools. Journal of Educational Computing Research, 1, 129-139.

Bork, A. (1987). The fourth revolution - computers and learning. In A. Bork (Ed.) Learning with personal computers. New York: Harper & Row. 3-18.

Brown, J. S., & Burton R. R. (1984). Diagnostic models for procedural bugs in basic mathematical skills. Instructional software: Principles of design and use. Belmont, CA: Wadsworth. 269-297.

Chaffin, J., Maxwell, B., & Thompson, B. (1985). Dragon Mix, Allen, TX: Developmental Learning Materials.

Clark R. E. (1985). Confounding in educational research. Journal of Educational Computing Research, 1, (2) 137-148.

Easley, J. A. Jr., & Zwoyer, R. E. (1975). Teaching by listening - Toward a new day in math classes. Contemporary Education, 47 19-25.

Herman, M. A., Kosel, M., & Sweedler, J. (1985). Code Quest. Pleasantville, NY: Sunburst Communications.

Kosel, M., & Carlson, J. (1984). The Incredible Laboratory, Pleasantville, NY: Sunburst Communications.

Kosel, M., & Fish, M. (1981). Elementary Volume 7 - Pre-reading/counting (731). St Paul, MN. MECC.

Kosel, M., & Fish, M. (1984) The Pond. Pleasantville, NY: Sunburst Communications.

Kosel, M., & Fish, M. (1983). Factory. Pleasantville, NY: Sunburst Communications.

Kulik, J. A., Kulik, C-L, C., & Bangert-Drowns (1985). Effectiveness of computer-based education in elementary schools. Computers in Human Behavior, 1, 59-74.

Neuman, D. (1986, November). Qualitative research: What can it bring to CBE? Paper presented at the 28th International Conference of the Association for the Development of Computer Based Instructional Systems, Washington, DC.

Sheingold, K., Hawkins, J.,& Char, C. (1984). "I'm the thinkist, you're the typist": The interaction of technology and the social life of classrooms. Journal of Social Issues, 40 (3), 49-61.

Siegel, M. A., & Davis, D. M. (1986). Understanding computer-based education. New York: Random House.

Snyder, T. (1983). Snooper Troops. Cambridge, MA: Spinnaker.

Trumbull, D. J. (1986). Games children play: A cautionary tale. Educational Leadership, 43 (2) 103-110.

Turkle, S. (1984). The second self: Computers and the human spirit. New York: Simon & Schuster.

Wang, M. C., & Lindvall, C. M. (1983). Individualdifferences and school learning environments. In E. W. Gordon (Ed.), Review of Research in Education, Vol 11, Washington D. C: American Educational Research Association.

Contact: Rosemary Sutton
College of Education
Cleveland State Univ.
Cleveland, OH 44115

Measurement of Spatial-Symbolic Information Processing Skills

Arthur L. White - The Ohio State University - Columbus

Donna F. Berlin - The Ohio State University - Newark

ABSTRACT

This paper describes the construction and validation of measures of spatial and symbolic processing skills derived from integration of technology into classroom instruction. The focus is on the ability of young children to recognize and extend designs, patterns, and sequences. Analysis of 79 third, fourth, and fifth graders' responses to 42 items revealed five factors: (1) Analogy, (2) Monotonic Progression, (3) Figural Pattern, (4) Rule Application, and (5) Attribute Identification-Rule Application.

A. Purpose. The purpose of this research is to develop an instrument to measure spatial and symbolic information processing skills related to the recognition, application, and understanding of designs, patterns, and sequences. This instrument is of special relevance to the study of influences that educational technology (e.g., overhead projectors, calculators, video players, and computers) may have on the abilities of children to recognize, extend, and/or construct designs, patterns, and sequences from a set of events. With the aid of technology these events are easily presented as geometric, figural, graphic, numeric, symbolic, and textual representations. Students can experience a variety of stimuli in different representational forms and perform tasks which vary in the degree of interaction they have with technological learning environments. The students can also interact with animated environments such as computer graphics and kinetic print. There is a need to better understand the role of visual processing of information in the learning of concepts, processes, and skills. More specifically, there is a need to better describe and understand the mechanisms whereby the various applications of technology influence learning. To accomplish these goals it is necessary to have precision instruments sensitive to the specific skills being targeted.

B. Theoretical Basis. According to McGee (1979), spatial ability is composed of two main factors: spatial visualization and spatial orientation. Spatial visualization involves the ability to manipulate and transform mentally two- and three-dimensional objects. Spatial orientation involves the ability to perceive the elements in a pattern, to compare patterns, to grasp changing orientation in space, and to determine the position of one's body in space.

The study of patterns or searching for regularities, hypothesizing or identifying rules, and inferring or predicting are essential skills that pervade the curriculum. Students need manipulative and exploratory activities to develop the spatial skills of visualization and orientation. Studies have concluded that spatial ability can be improved through training and instruction (Eastman & Carry, 1975; Hill & Oberhauf, 1979; Lord, 1985; Small & Morton, 1983; Smith & Litman, 1979; Talley, 1973). An instrument to evaluate these skills is suggested. The influence of technology on the development of spatial abilities is the focus of this research. The Spatial-Symbolic Processing Test will provide researchers with an instrument to facilitate study of the development of these skills and their relationship to instructional resources and design.

C. Procedures. In this study design, pattern, and sequence are conceptually defined as follows: (1) design - an arrangement of elements; (2) pattern - an arrangement of elements related by a qualitative rule (e.g., organizational, structural, spatial); and (3) sequence - an arrangement of consecutive elements related by a quantitative rule (e.g., numerical, temporal, ordinal). An instrument including 42 items was constructed to measure recognition and extension of designs, patterns, and sequences. These items include geometric figures, numbers, symbols, or words and require the subject to match designs and/or select or draw the next element in a pattern or sequence of elements. The items for the original instrument were grouped as shown in Table 1.

Table 1

Distribution of Items on Total Instrument

Item Groups	Number
Pattern Recognition	6
Geometric & Design Analogies	6
Geometric Patterns	8
Geometric Sequences	6
Word Order	6
Number Pattern & Sequence	10
Total	42

These items were administered to a sample consisting of 107 third, fourth, and fifth grade children from a rural Midwestern public school. The subjects were from the classrooms of six different teachers, two at each grade level. Due to absences, incomplete data, and inappropriate responses some of the subjects were dropped from the analysis. The resulting sample included those subjects with valid, complete data on all items. These 79 subjects were distributed by Grade Level and Gender as shown in Table 2.

Table 2

Grade Level by Gender Distribution of Final Sample

Gender	Grade Level			
	3rd	4th	5th	Total
Male	16	15	11	42
Female	12	14	11	37
Total	28	29	22	79

D. Data Analysis. The analyses of the responses included: 1) the computation of descriptive statistics with frequency distributions for each item; 2) principal components analysis for identification of underlying constructs; 3) computation of Cronbach's Alpha internal consistency reliability coefficients for the resulting principal components factors; 4) intercorrelations for all items, scales, Grade Level, and Gender; and 5) Grade Level by Gender group comparisons using a 2x3 factorial multivariate analysis of variance with the five subscales from the principal components analysis as dependent variables.

E. Results and Conclusions. The overall internal consistency reliability for the original instrument was computed based on the 79 subjects. The Cronbach Alpha reliability estimate from this calculation was 0.85.

A principal components analysis resulted in a six factor solution which accounted for 45.1% of the variance. Of these six factors, the first five were interpretable and resulted in clusters of items for five scales. The factor numbers, names, percent of variance accounted for by the factors (%Var), number of items with factor loadings of greater than 0.40 (k>.4), and the reliabilities (Alpha) for these five scales are given in Table 3.

Table 3

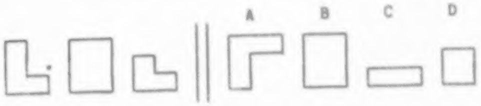
Principal Components Factors

Factor #	Name of Scale	% Var	k>.4	Alpha
1	Analogy (AS)	16.3	8	0.81
2	Monotonic Progression (MPS)	7.2	7	0.67
3	Figural Pattern (FPS)	6.1	5	0.76
4	Rule Application (RAS)	5.7	6	0.72
5	Attribute Identification (AI-RAS)	5.1	4	0.60

The responses to the items referred to in Table 3 were totaled for each factor to obtain subscale scores. These scales are described as follows:

Factor 1 - Analogy Scale (AS). This scale is comprised primarily of geometric analogy items. The subject was given two geometric figures which differed by some attribute. They were then given a third figure and asked to select the fourth from a set of four distractors. The attributes of the items and the cognitive processes involved in responding to the items for the AS include: analogy, comparison, two element pattern recognition, two element pattern extension, geometric representation, numeric representation, spatial form, orientation, rotation, shape, size, and shading. The following item had the greatest factor loading (.77) and is representative of the AS.

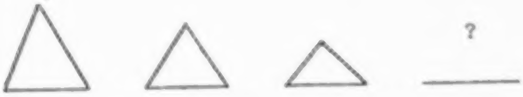
Which figure comes next?



Factor 2 - Monotonic Progression Scale (MPS). This scale includes numeric and geometric patterns based primarily upon the spatial organization or relationship involving a consistent increase or decrease in an attribute from element to element. For the most part the numeric patterns involve grouping or clustering of events or stimuli as opposed to numeric rules. The attributes of the items and the cognitive processes involved in responding to the items for the MPS include: pattern recognition, pattern extension, grouping, numeric representation, spatial form, height, increasing, decreasing, shape, distribution, and rotation. The following item had the greatest factor loading (0.59) on the MPS.

17, 18, 19, 82, 83, 84, □, 103, 104

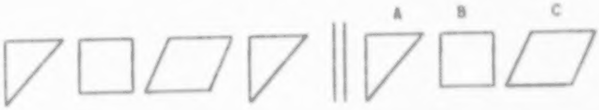
An example of an item using geometric representation for the MPS with a factor loading of just under (0.59) follows.



With items such as these the subject was given an assortment of nine triangles of various sizes and shapes from which to select the best response.

Factor 3 - Figural Pattern Scale (FPS). This scale includes items presenting geometric patterns and sequences. They consist of geometric representations with repeating patterns or a progression of changes. For the most part the progression of changes involves only one attribute at a time. The critical attributes include characteristics such as shape, size, or orientation. The attributes of the items and the cognitive processes involved in responding to the items for the FPS include: spatial form, geometric representation, multiple element pattern, counting sequences, number, temporal order, size, increasing, decreasing, pattern recognition, and pattern extension. The following item had the greatest factor loading (.82) and is representative of the FPS.

Which figure comes next?



Factor 4 - Rule Application Scale (RAS). This scale is comprised of items which require the subject to recognize sequences and apply a rule. These sequences generally progress from large to small or small to large. The attributes of the items and the cognitive processes involved in responding to the items for the RAS include: words, size, number, rule (addition or subtraction) identification, and rule application. The items are of two types: Type 1. (3 items) words which represent objects that can be put in order by size of the concept they represent. The item with the greatest loading (0.76) on the RAS follows.

Order from smallest to largest.

book, page, word

__________, __________, __________

Type 2. (3 items) number sequences which involve simple rules such as add 3 or subtract 5. The item with the greatest loading (0.69) for Type 2 items on the RAS follows.

9, 12, ☐, 18, 21, ☐

Factor 5 - Attribute Identification-Rule Application Scale (AI-RAS). This scale is comprised of items which require the subject to determine the continuum represented by two words and to select from a list of three the word which comes next. The attributes of the items and the cognitive processes involved in responding to the items for the AI-RAS include: words, attribute identification, rule identification, feature identification, pattern recognition, pattern extension, sequence recognition, sequence extension, temporal order, quantity, subsumer, and hierarchy. The item with the greatest loading (0.64) on the AI-RAS follows.

boy, teen, ___?___

Choices: baby, child, man

The following section is descriptive of the five scales from the principal components analysis and the relationships of these scale scores to other variables. Tables 4,5,6,7, and 8 give cell means for Grade Level by Gender, Grade Level, and Gender for each of the scales.

Table 4

Analogy Scale Means

	Grade Level			
Gender	3rd	4th	5th	Total
Male	4.88	5.73	5.27	5.29
Female	6.50	5.21	7.18	6.22
Total	5.57	5.48	6.23	5.72

Table 5

Monotonic Progression Scale Means

	Grade Level			
Gender	3rd	4th	5th	Total
Male	3.13	3.80	5.09	3.88
Female	3.67	3.64	3.55	3.62
Total	3.36	3.72	4.32	3.76

Table 6

Figural Pattern Scale Means

	Grade Level			
Gender	3rd	4th	5th	Total
Male	3.00	3.67	4.09	3.52
Female	3.50	3.79	3.36	3.57
Total	3.21	3.72	3.73	3.54

Table 7

Rule Application Scale Means

	Grade Level			
Gender	3rd	4th	5th	Total
Male	5.00	6.00	5.91	5.60
Female	6.50	5.43	7.36	6.35
Total	5.64	5.72	6.64	5.95

Table 8

Attribute Identification-Rule Application Scale Means

	Grade Level			
Gender	3rd	4th	5th	Total
Male	2.38	2.73	3.18	2.71
Female	3.50	3.29	3.55	3.43
Total	2.86	3.00	3.36	3.05

Correlations of these five scales with Grade Level and Gender were computed. These correlations are reported in Table 9.

Table 9

Correlation of Spatial-Symbolic Processing Test Scales with Grade Level and Gender (N=79)

Scales		AS	MPS	FPS	RAS	AI-RAS
Grade Level	r	.21*	.19	.13	.21*	.19
	p	.04	.07	.20	.04	.07
Gender	r	.11	-.11	-.03	.08	.31**
	p	.27	.28	.77	.43	.002

* $p < .05$
** $p \leq .01$

Grade Level was significantly correlated with the Analogy Scale ($r=0.21$, $p=.04$) and with the Rule Application Scale ($r=.21$, $p=.04$). In each instance the Grade Level accounts for only 4% of the variance in the scale score. These scales involved association or correlational reasoning. On both of these scales children at the higher grades tend to do better. Gender was significantly correlated with the Attribute Identification-Rule Application Scale ($r=0.31$, $p=.002$). The variance accounted for by Gender is 9.6% of the AI-RAS.

A Grade Level by Gender multivariate analysis of variance with the five Spatial-Symbolic Processing Test scales as dependent variables was computed. The results of this analysis and the follow-up univariate analyses of variance for the significant Gender effect are given in Tables 10 and 11.

Table 10

Multivariate Analysis of Variance of Spatial-Symbolic Processing Test Scales by Grade Level and Gender

Effect	Wilkes Lambda	F	p
Grade Level by Gender	.78	1.81	.06
Grade Level	.89	.82	.61
Gender	.81	3.27	.01**

** $p \leq .01$

Table 11

Univarate Analyses of Variance of Spatial-Symbolic Processing Test Scales (Gender Effect)

Scale	MS	F	p
AS	19.52	2.54	.12
MPS	2.89	.66	.42
FPS	.03	.01	.92
RAS	12.19	2.70	.11
AI-RAS	8.94	9.21	.003**

** $p \leq .01$

The multivariate Gender effect was significant ($p=.01$) with the girls doing better than the boys. The univariate analyses as well as the correlations indicate that the Attribute Identification-Rule Application Scale was the primary variable responsible for this effect. This scale required the subjects to interpret the meaning of a sequence of words, identify the relevant attribute, and identify a quantitative rule to extend the sequence to the next element. The girls ($X = 3.5$) were better than the boys ($X = 2.4$) on this task.

F. *Interpretations and Recommendations*. The original instrument was a collection of items to determine how children perceive and process spatial and symbolic information. The instrument included designs, patterns, and sequences displayed as geometric shapes, numbers, and words. The subjects were asked to recognize or extend sets of events.

The principal components analysis produced subscales with more specific constructs than were perceived in the design of the original instrument. The cognitive processes and skills which were revealed by this analysis include:

Factor 1 - Analogy. These items require a recognition of relationship between two events so as to apply it to a second pair of events.

Factor 2 - Monotonic Progression. These items require recognition of the relationship among more than two events. The rule is either an increasing trend or decreasing trend from event to event.

Factor 3 - Figural Pattern. These items require the recognition of repetition of events so as to predict the next event.

Factor 4 - Rule Application. For these items the attribute of focus is either given or easily discernible and the task is to determine the rule and apply it.

Factor 5 - Attribute Identification-Rule Application. For these items the task is to identify the relevant attribute then identify the rule and apply it.

On these five scales, the girls did better than the boys on the tasks which require a higher level of information processing (e.g., analogy, attribute identification, rule identification, rule application). The rule identification and application involve numbers and concepts and continuums represented by words. There were no differences on the tasks involving increasing or decreasing patterns and no differences between boys and girls in recognition of patterns of geometric shapes.

These scales will be expanded by adding items similar to those clusters identified by this study. The influence of technological instructional activities involving geometric, numeric, and textual stimuli can better be explored. These scales will provide a sensitive measure to study the benefits which technology provides for learning.

References

Eastman, P. M., & Carry, L. R. (1975). Interaction of spatial visualization and general reasoning abilities with instructional treatment in quadratic inequalities: A further investigation. Journal for Research in Mathematics Education, 6(3), 142-149.

Hill, D. M., & Oberhauf, P. A. (1979). Spatial visualization, problem solving, and cognitive development in freshmen teacher education students. Science Education, 63(5), 665-670.

Lord, T. P. (1985). Enhancing the visuo-spatial aptitude of students. Journal of Research in Science Teaching, 22(5), 395-405.

McGee, M.G. (1979). Human spatial abilities. Sources of sex differences. New York: Praeger Publishers.

Small, M.Y., & Morton, M.E. (1983). Spatial visualization training improves performance in organic chemistry. Journal of College Science Teaching, Sept./Oct., 41-43.

Smith, W. S., & Litman, C. I. (1979). Early adolescent girls' and boys' learning of spatial visualization skill. Science Education, 63, 61-66.

Tally, L. H. (1973). The use of three-dimensional visualization as a moderator in the higher cognitive learning of concepts in college level chemistry. Journal of Research in Science Teaching, 10, 263-269.

Dr. Arthur L. White
5456 Foxhound Lane
Westerville, OH 43081
Ph. 614-890-2453

TOWARD AN IDEAL VISUAL INTERFACE FOR COMPUTER-BASED AUTHORING SYSTEMS

Copyright (C) 1987, All Rights Reserved

By Dan G. Friedrich
Friedrich, Klatt and Associates

ABSTRACT

An ideal computer-based training (CBT) system would let programmer and non-programmer authors alike quickly create both simple CBT programs and the most sophisticated programs an experienced programmer could create with a traditional language. In sketching out an intuitive, visual CBT user interface that might realize these goals, we can learn from computer software tools such as structured programming languages, pseudo-code, metalanguages, spreadsheets, outlines, graphics, and, especially, flow charts. The visual model that emerges, especially if tightly integrated with traditional text-based programming, would let authors design more and code less. It would let them design more by flexibly highlighting program flow of control, hierarchical module structure, and other aspects of the "big picture." It would let them code less by reducing both the physical and imaginative distances they must travel in moving between "code" and finished product.

I. PROGRAMMING'S PHYSICAL AND IMAGINATIVE MOVEMENTS

With traditional programming languages, the major challenge of the coding is in moving between source code and finished product. This movement is partly physical--one edits source code, exits the editor to test, returns to the editor, and so on. But it is also imaginative--one must *conceive* of the connections between these two necessary representations of the work.

Generally, one can only reduce one of these demands by increasing the other. To reduce the physical mode changes, one works for longer stretches on the source code before testing. But this requires greater imaginative gymnastics because one must imagine a longer code-to-product script. To reduce the imaginative mode changes, one modifies code in tiny increments, testing immediately, which requires more physical movements.

Of course, we can make these movements more conveniently with tools such as memory-resident editors that pop in and out of view with a keystroke, and compilers that respond to errors by automatically bringing up an editor and placing the cursor on the offending code. And interpreters can cut out the compile step altogether (usually at the expense of speed and source code protection). But the only way to significantly reduce the physical and imaginative work in the traditional programming model is to "pay your dues," to acquire expertise by developing over time the mental templates or intellectual tools that any expert acquires.

As we shall see, a visual approach to programming can reduce both these movements. In designing this ideal visual interface or "language" for CBT, we can learn from computer software tools such as structured programming languages, pseudo-code, metalanguages, spreadsheets, outlines, graphics, and, especially, flow charts.

II. LESSONS FROM EXISTING COMPUTER SOFTWARE TOOLS

A. FREE TO BE STRUCTURED: STRUCTURED PROGRAMMING

Structured programming languages such as C, Pascal, and Modula-2 established programming principles that we should apply to our visual interface. For example, they demonstrated the importance of breaking programs into modules that can be nested within one another, with the associated notion of variables whose scope or "existence" is limited to a particular module and its subordinate modules. These languages also popularized the related concept of variables passed as parameters to other program modules whose behavior varies with these inputs. Finally, they gave us ways to represent these structural relationships in a two-dimensional space with their conventions for declaring variables at the beginning of programs, for explicitly naming and separating modules, and for indenting lines of code to show hierarchical relationships.

Through the success of these principles, structured programming has demonstrated that it is easier to write and maintain programs guaranteed by the very definition of their language to be structured in certain ways; such programs simply cannot arbitrarily hop here and there, jumping bean style. And structured programming shows that our programming languages can require such structure, yet be flexible enough to implement any logically coherent program. Our visual interface ought to achieve the same balance.

B. ON NESTING BEHAVIOR: OUTLINE PROGRAMS

Just as structured programs visually represent hierarchical nesting of programming routines and subroutines, outlines visually represent the hierarchical relationships of general and specific points. Our visual language should incorporate the outline's powerful hierarchical aspects, as well as its elegant visual representation of these relationships through indenting lines, numbering sections, and emphasizing higher levels by underlining, capitalizing, or otherwise highlighting them.

A number of microcomputer word processing and other packages now offer electronic versions of outlining, including at least one--Framework[1]--that lets one organize programs in nested outline frames. Our visual interface should borrow from these software packages the idea of collapsing sections of a program outline, as well as the notion that when we select, copy, move, or delete an element at a higher level, we intend also to perform the same action on subordinate elements.

C. MY PSEUDO-CODE'S OK, YOUR PSEUDO-CODE'S OK

In actual practice, many programmers, if they use any tools to represent programming tasks, probably use pseudo-code. This private or personal code captures the key points in a program with statements that are terse approximations of actual program code. (Of course, if the language is C, the pseudo-code might be a full, readable approximation of terse *actual* code!) An example of pseudo-code is shown in *Figure 1.*

```
Show welcome screen
Explain concept
Ask question 1 about concept
Get user response
If answer is "yes"
   Ask question 2
If answer is not "yes"
   Show error message
   Re-ask question 1
...
```

Figure 1. Pseudo-code example.

Such "code" need not conform to any programming language rules; indeed, it might be implemented in many languages. It may not even be internally consistent. It simply states those aspects of the program design the programmer considers worth noting.

Pseudo-code demonstrates the advantages of looking at a program design independent of any particular implementation, of focusing flexibly on the flow of control and other key program points, of looking at our programs at various levels of abstraction. The very *ad hoc* nature of pseudo-code "languages" gives their users the freedom to look at those aspects of a program that are interesting, complex, problematic, unclear, incomplete, or critical.

Our ideal visual programming interface should give the same freedom to its users to concentrate on just those things they are interested in. This means allowing them to enter major module names and relationships without filling in details until later, to temporarily hide details they *have* entered, and to simultaneously view different parts of their programs at different levels of abstraction. These would offer the visual equivalents of our pseudo-code author expanding "Ask question 2" *(Figure 1)* into "Show prompt for question 2" and "Get user response," then hiding these details "behind" the original question. To offer this flexible abstracting, the visual system must "understand" the programming structures while they are being edited and not, *a la* traditional programming systems, simply consider them arbitrary strings of symbols until they are interpreted or compiled.

D. TALKING ABOUT TALKING: METALANGUAGES

Unlike pseudo-code, which describes *programs* that might be implemented in many programming languages, metalanguages are formally defined languages with strict syntax rules, used to describe particular programming *languages* apart from any program that might be written with them. A metalanguage is a "super" language that we use to describe the formal properties of ordinary languages. For example, in Backus Naur Form (BNF) one defines a programming language through a series of interdependent, collectively exhaustive "productions" such as "<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9", which says that, for the given programming language, a "digit" may be 0 or 1 or 2 and so on.

Despite their differences, both pseudo-code and metalanguages demonstrate that their reductionist magic can define the fundamental functions of their respective domains--programs and programming languages. In the same ways, our visual language should both allow authors to create the visual equivalent of pseudo-code in designing *programs* and be logically coherent enough to permit us to formally define it with a BNF-like metalanguage with a visual vocabulary.

Further, our visual language ought to have the programming capabilities represented by the BNF productions of leading general-purpose programming languages. This will go a long way to insure that it has all of the primitive building blocks needed to build any complex program, including a variety of data types, input and output functions, looping and branching capabilities, calculating and comparing operations, and the ability to assign values to variables.

E. ELECTRONIC PLAY DOUGH: SPREADSHEETS

We can also learn from the ledger sheets of our age, spreadsheet programs. These electronic grids are in principle like play dough; one can place text, numbers, and formulas anywhere on their two-dimensional row-column visual spaces. In practice, of course, we lay entries out logically, placing data in tables, subtotals at the bottom of columns, and so on *(Figure 2)*.

---

[1] Framework is a registered trademark of Ashton-Tate.

```
       A      B     C     D       E              F
1     Jan    Feb   Mar  Total  Total Formula  Algebra
2    -----  ----- ----  -----  -------------  ---------
3    100.0  200.0  50.0  350.0  +A3+B3+C3      X=A+B+C
```

Figure 2. Spreadsheet example.

Spreadsheets have shown that we can translate an algebraic expression into a visual map with constants and formulas placed at (x,y) coordinates (the cell addresses at the intersections of columns and rows). The "variables" in the formulas are addresses of other cells on the grid. (Spreadsheets also typically let one assign names to these addresses and then use these names as the variables.) When the spreadsheet calculates a formula, it uses the values in the cells referenced by the cell addresses. Of course, these cells may in turn reference other cells, which reference others, and so on.

Sophisticated as such spreadsheets are, they are essentially snapshots, static pictures, while our visual interface must represent a dynamic process--program flow. Even so, we can borrow from spreadsheets the notion that formulas and variables may be placed at spacial locations and be linked to other formulas and variables in chains of dependencies. We also learn that our visual language must allow us to flexibly lay out program elements, and to manipulate both individual elements (akin to cell addresses) and groups of elements (ranges), selecting them by name or cursor pointing.

The newest three-dimensional (3D) spreadsheets, which add a "z" dimension to the familiar (x,y) coordinates, offer yet another lesson, namely that our ideal visual interface should offer the power of depth, of showing details or parallel elements hiding "behind" as well as "below" or "beside" elements currently visible on the screen.

F. BIG CHEESE, LITTLE CHEESE: SCALE IN GRAPHICS

Graphics programs are more structured than spreadsheets; they require that their elements be spacially related to each other in fixed ways. However, because they only insist on *relative*, and not *absolute* spacial distances, graphs and drawings can be equally well represented in different scales. Our visual interface should offer a similar scaling capability.

Increasingly, graphics programs offer 3D graphics, which use perspective and shading or color to simulate on the two-dimensional screen or paper a visual 3D object. (This goes beyond the three-dimensionality of 3D spreadsheets, which typically make no attempt to simulate *visually* their three conceptual dimensions.) Naturally, our ideal visual interface should offer this feature too--showing, for instance, subordinate modules *visibly* "behind" foreground parent modules.

G. OF FLOW-CHARTS AND FORGIVENESS

Many programmers who sketch their programs in their heads or, at best, with a bit of pseudo-code, know they should use flow charts. They know that flow charts nicely represent program flow in a visual and intuitive way. Our ideal visual interface should acknowledge this by finally moving flow charts to the very heart of programming, creating programs in which the flow chart *is* the program.

Like pseudo-code, flow charts let us focus on the fundamentals of a program, but they do so in a strikingly visual way. Nearly all the common symbols and elements of flow charts should be incorporated into our ideal visual interface, including the following:

Start/Stop. These simple but essential program elements, symbolized by rectangles with rounded corners, mark the only begin and end points of programs.

Module. This program element, depicted by a rectangle, represents a program unit. Depending on our purposes, we might define modules as separate source code files or as procedures, which might or might not be in separate files. Or we might define modules as groups of instructions that execute sequentially from start to finish, with one entry point and one exit point. An example might be the instructions inside a simple loop. Alternatively, we might define modules more broadly and flexibly as any contiguous lines of code that we wished for the purposes at hand to consider as units, from single lines of code to entire programs. Regardless of how we define the modules, our visual interface should let us treat them as "black boxes" whose contents we may temporarily hide or ignore to focus on their functions and relations to other modules.

Conditional Branch. This diamond-shaped element represents forks in the programming road based on "true or false" logical tests, such as whether the value in the variable "LastName" is "Smith". If the condition tested is true, the program branches one way; if false, it branches another. Our visual interface should display inside these diamonds as much of the conditional expression as possible, allowing authors to zoom into them to examine or edit the full expressions.

Flow of control. These lines or arrows show the paths the program can take through all the program modules, branches, and other elements they connect--clearly a necessary feature of our visual interface. The "control" refers to that dynamic position in the code that has control or is executing at any moment. If the flow is symbolized by a line, the direction will be generally from left to right and from top to bottom, and in any case should be clear from the context. If the flow is symbolized by an arrow, the direction will be (surprise!) in the direction of the arrow.

Of course the greatest benefit of making flow charts integral to programming and compelling all of us to use them, is to relieve the guilt of all those programmers who, until now, never did.

III. TRADITIONAL PROGRAMMING WITH VISUAL PROGRAMMING?

In moving from the theoretical promise of visual programming to concrete designs, we face many serious issues, not least of which is whether our imagined visual system should act as a complete run-time interpreter, storing its "programs" in a special file format that only this interpreter could save and retrieve. Alternatively, it might have a companion traditional programming language. Such a language could be integrated into our visual system in at least the following ways:

A. TRADITIONAL CODE FOR NASTY BUSINESS

The textual language could be used to program particularly nasty modules that one could not implement with the visual system. These limitations might be due to a failure to implement the visual system in such a way that it could handle all major classes of programming tasks. Or they might be due to conscious decisions to leave out functions better performed with traditional languages. Either way, since we are after an *ideal* visual interface, we can set this band-aid solution aside until design problems force us to resort to it.

B. TRADITIONAL CODE FOR CHOOSY AUTHORS

In this case, the visual interface would be able in principle to handle all programming tasks, but the user could optionally hook in modules created with a traditional language. Users might do so because they were experienced and efficient in the traditional programming skills and wanted to keep using them for some purposes. Or they might already have libraries of effective modules. Since this flexibility would be a welcome addition to our visual interface, we should support it, perhaps by making it possible to load into memory, and then call, object code subroutines created with a traditional language. Or perhaps we could offer our courseware system as a library of modules to link in with other code from a traditional language. In any case, for maximum flexibility, our visual system should be able to remain in memory while running any independently executable program, which of course might have been written in any language.

C. TRADITIONAL CODE AS LEAVES ON THE VISUAL TREE

In this approach, all branches on the visual interface lead ultimately to little leaves of traditional code (or at least stored keystrokes or macros). The visual interface would be the big control program in the sky, handling flow of control and parameter-passing between its little packets of traditional code, but not itself performing any other functions. Of course, we would have to decide where to draw the line between the visual interface and the traditional code. Candidates would include all the possible boundaries of modules that we discussed earlier.

This approach makes sense, particularly if our visual interface sits, as it should, upon a rich interactive CBT programming environment with built-in tools such as screen painters, windowing, music generation, a variety of pop-up menu styles, and other sophisticated ways of getting input from users. (Of course these tools should also be highly visual. For more on the specific features a CBT system should have, and how they might be implemented, see my paper, "Design Issues in Developing Computer-Based Authoring Systems."[2])

Our visual interface could tap these features by representing them as icons with associated fill-in-the-blank settings sheets that determined exactly what they did in each instance--which screen to pull up, what to show on the menu, and so on. Of course, some features of this environment would be easiest to use by simply storing keystrokes needed to activate them interactively. Features suited to this macro method might include changing the color of existing text on the screen, writing short prompts or error messages, creating animation effects, and so on.

D. TRADITIONAL CODE FROM A VISUAL "FRONT END"

We could integrate the visual interface even more closely with traditional code by having the interface act as a sophisticated version of traditional code generators, which generate code based on instructions issued by the programmer, often through menus. Our generated programs could be interpreted directly by the visual system or compiled in traditional ways to create the actual run-time CBT program.

The main challenge here is devising ways for authors to repeatedly hone their programs using the visual interface, which in turn means the visual system must be able to reconstruct and re-save its programs. Because of the difficulties in interpreting ordinary source code, code generators typically produce simultaneously both the editable source code and a special internal version used to reconstruct the program. But when, as inevitably happens, authors bump against a limitation in the generator and modify the generated source code directly, they cut themselves adrift from the generator; from that point on, they either code without the generator or regenerate and thereby wipe out the changes they made to the source code.

Ideally, then, we might have a source code generator that was also a source code reader, foregoing the pristine internal version. To create this, we would have to not only overcome the sheer difficulty of interpreting this source code with all of its potential syntax errors but also somehow store and read special information on the exact configuration of the most recent visual state. (Without such special information we would have to reconstruct this state based only on default layout patterns. We would be unable to allow the flexibility in this area that we have articulated as an important design goal.) One solution is suggested by QuickSilver[3], a dBASE III

---

[2] Published in the proceedings of the Society for Applied Learning Technology's third annual conference on the "Development of Effective Interactive Instruction Materials," June 17-19, 1987. Also available from the author.

[3] Quicksilver is a registered trademark of Wordtech Systems.

Plus[4]-compatible compiler that has statements not found in dBASE, yet allows its source code to remain executable under the dBASE interpreter by permitting one to embed these special commands in statements that look to dBASE like non-executable comments.

This fourth approach's close integration with traditional languages offers a number of advantages over looser meshes. It would allow authors with programming skills to continue to use both those skills and existing libraries of code. It would give authors the flexibility to program some modules in traditional ways while programming others with the visual interface. For example, they could program the highest levels of programs with the visual interface and fill in the details by writing code directly. In short, they could mix these methods as they wished. Furthermore, as we enhanced the visual CBT system to support additional source languages, including those outside CBT, we would be building a potentially universal visual interface for general-purpose programming that built upon existing languages rather than leaving them behind. It would let its programmers use new languages productively without necessarily even learning them, or, paradoxically, use them productively as a *means* of learning them.

IV. CONCLUSION: TOWARD A PERFECT VISUAL LANGUAGE

Our ideal visual CBT programming environment has begun to take shape. It looks on the surface most like traditional flow charts, with symbols for program start and stop points, modules, conditional branches, and flow of control. But it also shows its grounding in the other computer software tools we examined.

From these we learn that our visual language must be formally definable, yet flexible enough to permit a variety of programming styles and to accomplish any theoretically possible CBT programming task. It should move beyond a two-dimensional conception of the programming "space" to one that is both conceptually and visually three-dimensional, in which one can "zoom" in and out of levels of detail. To assist in this focusing, we must make it especially easy to highlight those parts of programs that programmers are typically most interested in--the flow of control points, particularly between major program "modules." For any of this to be possible, the program must "understand" the program elements being created, rather than treating them as visual equivalents of arbitrary lines of source code in a text editor.

In short, the visual approach to CBT sketched here can reduce the difficulty of programming as traditionally practiced, even by experts. It does so first by reducing the physical distance between "code" and product by making it unnecessary to refer to pseudo-code or flow charts on paper, and by offering an interactive environment in which running the "program" is only a keystroke away (and fast, too, since the system already knows about the program elements and their relation to each other). Second, it reduces the imaginative movement, the conceptual work, by presenting program elements and logic more intuitively than text can. Finally, in those limited cases when a strictly visual interface did not successfully reduce these movements, a visual system that both generated and read traditional code would let authors use traditional coding as they wished.

This reduction in programming workload is a loss if one values the mental games or puzzles of traditional programming, a gain if one prefers to be freed to concentrate fully on substantive design tasks. Certainly the focus on substantive tasks is a bonus for CBT, which must enlist non-programmers. In effect, we can collapse the traditional CBT division of labor between "writer" and "programmer", leaving both tasks to a single *author*.

V. ABOUT THE AUTHOR

DAN G. FRIEDRICH is president of Friedrich, Klatt and Associates, a microcomputer consulting firm based in Chicago. He does computer training, consulting, and programming for a wide variety of organizations. His software products developed or under development for commercial resale include a legal office management system; an insurance coverage information system provided to hospitals by hospital associations in a number of states; a code generator for business database applications; an interactive help system for IBM PC's and compatibles; a CBT program teaching advanced writing skills; and a PC-based courseware authoring system that is the basis for this paper.

He has a bachelor's degree from Valparaiso University and was a doctoral student in political science at the University of Chicago until computer-based distractions took him down his present path. His other distractions include writing and acting in comic skits in which he impersonates politicians and celebrities. His address is:

Dan G. Friedrich, President
Friedrich, Klatt and Associates
5344 S. Woodlawn Avenue
Chicago, Illinois 60615
(312) 288-4849

---

[4] dBASE III Plus is a registered trademark of Ashton-Tate.

COMPUTERIZED ENVIRONMENT FOR INSTRUCTIONAL DESIGN :

Developing CBT with Artificial Intelligence based Workstation

Avshalom Aderet, Ph. D.

ELRON Electronic Industries Ltd.

INTRODUCTION

Although AI applications, particularly expert systems, have proliferated into many areas in recent years, the penetration of AI into education has hitherto proved very slow and rather disappointing. This is true for Computer Based Training (CBT) in general and computer courseware development in particular.

Market surveys over the last 5 years indicate a dramatic increase in demand for CBT for computer systems. Among the reasons for this are the expansion of computer systems; the fact that these systems are becoming more and more complicated; the tough competition among computer vendors; and user demand for better support. Training departments and customer support divisions are under increasing pressure to provide extensive, effective and efficient training.

THE PAUCITY OF CURRENT CBT DEVELOPMENT TOOLS

Although CBT has convincingly demonstrated its ability to provide superior teaching efficiency,existing CBT development tools do not adequately meet the challenge in real training environments. This somewhat surprising state of affairs is even more critical in the area of computer systems, where the demands are much tougher. Systems change rapidly, new products hit the market almost daily - and all these systems have to be taught. And although CBT appeals to a heterogeneous and constantly changing group of users, the development procedures of existing CBT tools are far too rigid (Merrill, 1985). CBT has proved costly to develop and difficult to update.

PROBLEMS OF CBT DEVELOPMENT

Of all the elements in the CBT development process, the design and programming phases have proved the most inefficient and time consuming. CBT development involves manual designing, whereby courseware design is entirely off-computer, somewhat akin to electrical engineering technology before the advent of CAD/CAM.

AI has remained indifferent to the needs of the course designer, who receives no computerized, technical or didactic support.

Course programming can only be implemented by a professional programmer, and involves the use of either a programming language or an authoring system. Every minor modification in the course flow requires reprogramming, which in turn makes the entire development process combersome, tedious and expensive.

CBT design has hitherto been based mainly on artistic intuition and common sense. Automating this process involves overcoming conceptual barriers and practical problems that hamper the entire design process. It is thus hardly surprising that relatively few applications currently exist in this field (Aderet, 1986)

The automation of CBT design presents the AI community with a major challenge.

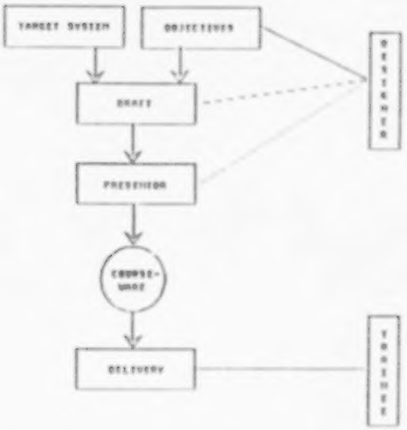
THE ACE SYSTEM

ACE (automatic Courseware Expert) is a new system that introduces a high degree of automation into CBT courseware design for training users of computer applications. Automating the CBT design process involves several underlying principles. Firstly, the automatic process generates a comprehensive model which integrates all the CBT elements into a manageable system. Next the automatic process generates/synthesizes those aspects of CBT that are mechanical and/or governed by formalizable rules such as feedback messages, screen layout, etc. Human intervention is required only to overrule and modify the automatically generated output and to complete those sections (e.g. metaphors, analogies, etc.) which are beyong the capabilities of the computer

The ACE system incorporates a variety of areas and disciplines: cognitive psychology, learning theory, man-machine interface design, computer technology and software applications. ACE is not designed to replace the courseware designer, but rather to increase the productivity of expert CBT writers and less experienced personnel.

CBT development with ACE involves the activation of 4 Major expert modules, as shown in Figure 1.

FIGURE 1 CBT DEVELOPMENT WITH ACE



(1) "Objectives" - specifying the training requirements for the course (e.g. type of lesson, characteristics of audience)
(2) "Target System" - describing the computer application for which CBT is intended.
(3) "Draft" - developing the course structure and all its didactic derivatives, e.g. trainee feedback instructions, course branching etc.
(4) "Presentor" - Developing the screen layouts for the entire course.

Since the ultimate goal of ACE is to generate courseware for a specific computer application, ACE involves the simultaneous implementation of two processes:

(i) The CBT requirements for the particular courseware being developed are specified in terms of the didactic model of ACE - "Objectives".
(ii) The specific computer application is mapped into the model of the domain (a generic type of application) already built into ACE - "Target System".

Following the activation of these two processes, the "Draft" is activated to generate the course structure and the general flow of the lesson. In order to develop the screen layouts, another automatic module, the "Presentor", is activated.

THE PRESENTOR

A major problem in CBT development is the actual generation of all the screen layouts. To illustrate the power of ACE as an integrated AI - based workstation, the Presentor expert module will now be described in greater detail.

The presentor is an expert system responsible for the final presentation of the courseware on the computer screen. A typical presentation of a screen is shown in Figure 2.

FIGURE 2 A TYPICAL PRESENTATION OF A LESSON SCREEN

The course presentation can significantly affect the length of the course; the number of interactions with the trainee; and most importantly - the overall acceptability of the course. Moreover, although it is relatively easy to formulate presentation decisions, it is often difficult and time-consuming to implement them with existing tools such as an authoring system. The automation of the course presentation process should therefore lead to a radical reduction in development time.

The use of automatic procedures means that changes in the presentation of the entire course can be implemented by merely changing a few parameters. In the ACE system, the Presentor is responsible for this automatic procedure.

CBT courses (or lessons) generally consist of a series of "frames", whereby the trainee progresses from one frame to another. This progression is controlled by user responses to system requests. A frame includes one or more display screens. Typically, a display screen of a tutorial course contains several "windows", each displaying a particular piece of information: a picture or form which simulates the screen of the application to be trained; a textual explanation of that picture; a command (what to do next); feedback on a wrong answer; and a graphic animation explaining a particular point.

The input of the Presentor basically consists of the CONTENT of the windows which are designed to be shown concurrently. The Presentor makes two types of decisions:

(I) Decisions concerning the overall layout of the screen, including the size and position of each window.

(II) Decisions concerning the presentation within each window, including formatting, margins and spacing, highlighting and colors, etc.

The Presentor's task is made more complex by the fact that these two types of decisions are NOT independent. For example, formatting decisions affect the size and position of the windows, while the amount of space allocated to a text window may effect its internal format. The number of presentation possibilities is consequently very large. The presentation itself is affected by many different parameters and constraints, such as the color of highlighting, the margins and the line-spacing in a text window, the maximum or minimum widths of windows, and more.

Architecturally, the Presentor is a mix between a classic expert system using a knowledge-base and rule-interpreter, and heuristic algorithms which solve some of the placement and formatting problems.

The implementation language is PROLOG, which has proved highly suitable for implementing such a mix of rule-based and procedural knowledge (Gudes, Shafrir & Gerlitz, 1986)

The power and flexibility of the Presentor significantly simplifies the task of screen design for courseware development, and provides consistency of presentation both within a course and across courses. Since the Presentor uses many global parameters, changing these parameters can change the presentation of all or part of the course. This allows the designer to "test" different presentation options, which is very difficult to achieve with existing authoring tools.

SUMMARY

As modern technology becomes more sophisticated, the need for technological training has become more acute. One of the most effective means of providing mass technological training is Computer Based Training (CBT). However, while it is widely recognized that CBT has a huge potential, this is far from being realized. ACE presents a new approach in CBT design and development. As illustrated by the Presentor, ACE provides a powerful environment for CBT design. It frees the designer from tedious and time-consuming tasks such as screen presentation, and makes CBT design work more productive and more rewarding.

AI has proved an innovative and powerful solution to the long-standing problem of developing technological training on a large scale basis. The ACE system successfully demonstrates these capabilities.

REFERENCES

1. Aderet, A. Will AI Improve CBT Development Process. A paper presented in the Conference on Applications of Artificial Intelligence and Expert Systems, Virginia, October 1986.

2. Gudes, E., Shafrir, A., Gerlitz, S., Software and Knowledge Engineering Methods in Buliding Expert Systems, Proceedings of the 2nd International Conference on Expert Systems, London, May, 1986

3. Merrill, M.D., Where is the Authoring in Authoring Systems?, Journal of Computer-based Instruction, Autumn, 1985.

Contact: Avshalom Aderet
Elron Electronic Industries Ltd.
POB 383
Or-Yehuda 60200
Israel

USING A WORKBOOK/ON-LINE TUTORIAL AS A STARTING POINT FOR CBT DEVELOPMENT WITH AN AUTHORING LANGUAGE

Martha R. Atwater*

Regency Systems, Inc.

Learning CBT techniques and a new authoring language can sometimes be overwhelming for courseware authors and programmers. Often the necessary learning time is too costly, and developers are forced to use a menu-driven authoring system for an application which really requires the power and flexibility of a more sophisticated authoring language.

Regency Systems, Inc., has developed a new workbook/on-line approach for teaching its TEL Authoring Language. The tutorial was designed to provide a quick way to "ease" into courseware development with TEL.

In approximately 18 hours of study time, the user becomes familiar with the basics of courseware planning and with TEL's most commonly used commands and functions. He/she becomes comfortable with TEL's editors, utilities and documentation and uses them to create a model lesson.

The presentation will use sample course materials to illustrate how written exercises, simulations and guided experiences combine to provide a starting point for developing courseware with TEL.

*Regency Systems, Inc.
3200 Farber Drive
Champaign, IL 61821
(217) 398-8067

THE DEVELOPMENT OF AN INSTRUCTIONAL PLANNER FOR AN INTELLIGENT TUTORING SYSTEM

Melissa Berkowitz,* David Emme, Stuart Macmillan

FMC Corporation

The purpose of this presentation is to report on the progress of a research program which is developing an Instructional Planner for an Intelligent Tutoring System (ITS). ITSs contain four principal components: a Student Interface, an Expert Model, a Student Model and a Pedagogy Model.

The Student Interface supports the display of information to the student and allows the student to respond to the system. The Expert Model contains the facts and procedures about the domain of interest. It supports the explanations, problems and responses given to the student. The Student Model contains a representation of the student's level of competence. The Pedagogy Model contains the knowledge of how to teach. It supports the tutoring knowledge of the system by making decisions about the sequence of instruction. Instructional planning, the focus of this effort, may be viewed as a function of the Pedagogy Model. Like human planners, machine instructional planners must respond to individual students several ways. It must identify the student's current state of understanding, determine what to teach next and provide appropriate feedback about the student's progress.

This presentation will examine the desired capabilities of an Instructional Planner and discuss the approach being pursued to create a Planner with these capabilities. The challenge of implementing such a system for a military training application will be also discussed.

*FMC Corporation
Central Engineering Laboratories
Artificial Intelligence Center
1205 Coleman Avenue, Box 950
Santa Clara, CA 95052

(408) 289-0462

3 2

COMPUTER-GENERATED VIDEO SLIDES AND STORYBOARDS:

ALTERNATIVES TO COMPUTER-BASED TRAINING

David W. Dalton,* David Goodrum

Indiana University

Although a number of powerful authoring environments exist for producing elaborate CBT lessons, these systems are often difficult for novice CBT authors to learn, and are therefore barriers to using computers for simple and straightforward lesson content.

This presentation describes two simple to use alternatives to such complex authoring environments. First, a method will be described whereby simple graphics created through "painting" programs such as PC Paintbrush or MousePaint can be easily linked together into an electronic "slide show". Once the images are created, the sequence and timing of the slides can be easily edited. The edited slideshow can then be displayed on the microcomputer to one or many learners or exported directly to videotape. Along with serving as stand-alone CBT lessons, this type of slide show can also provide developers with an ideal format in which to experiment with screen and lesson design.

Second, programs like PC Storyboard, designed for exactly this type of slide show can be used to create similar effects. In addition to the basic slide show concept, Storyboard offers more advanced features including response acceptance, variable font sizes and styles and simple animation.

Both of these systems are easy to learn and operate and serve as painless starting points for would-be CBT authors and developers.

*School of Education 210
Indiana University
Bloomington, IN 47405
(812) 335-6118

AUTHORING SYSTEM EVALUATION STRATEGIES

Daniel V. Eastmond*

Chevron Information Technology Company

Choosing the authoring system that meets the requirements of an organization is challenging, given the variety of instructional settings and the number of competing products. This session presents the evaluation strategy followed by the Chevron Information Technology Company (CITC) in their selection of an authoring system. The objectives and process of the several stages of evaluation will be detailed.

Upon completion of this presentation attendees will be able to: recognize the various organizational requirements that impinge on authoring system selection; identify crucial standards for narrowing the scope of the evaluation effort; explain the process of developing criteria for an "ideal" authoring system that meets their environment; define benchmarking and discuss its role in selecting a final candidate.

Samples of CITC's evaluation forms will be given to the attendees.

*Chevron Information Technology Company
P.O. Box 5032
San Ramon, CA 94583-0932
(415) 842-1908

ADAPTING OBJECT-ORIENTED DATABASE SYSTEMS
FOR USE AS AUTOMATED JOB-AIDS

Shirley Gresham, Leslie Ferguson, Brockenbrough S. Allen*

San Diego State University

Job-aids significantly reduce the need for technical training in a wide range of occupations and professions. Object-oriented database systems can be adapted to serve as computer-based automated job-aids. They encompass functions that link graphic elements in one display with superordinate, subordinate, or elaborative information in other displays. This additional information itself may be graphic, making it possible to develop custom job-aids with complex branching structures at low cost and with little programming experience. Object-oriented database systems can be used to develop heuristic aids as well as algorithmic aids and can accommodate a variety of job-aid formats. These database systems permit rapid access to large numbers of graphic displays in a way that help end-users to map spacial, conceptual, procedural and causal relationships between database structures.

This presentation describes major features and capabilities of object-oriented database systems and will overview issues involved in constructing automated job-aids. It will demonstrate how an object-oriented database can be adapted for job-aid development and it will simulate an automated job-aid as it might be used to assist customer service representatives with technical questions phoned in from the field.

*Educational Technology
San Diego State University
San Diego, Ca 92182-0311
(619) 265-5439

SOFTWARE TOOLS FOR TRAINING DEVELOPERS

Greg Kearsley*

Park Row Software

With the exception of authoring systems, there has been little automation of the instructional development process. Analysis, design and evaluation activities are done almost completely by hand. This makes the development process very labor intensive and subject to human performance limitations and skill differences. To reduce the time and costs involved in developing instruction, as well as improving reliability, we need to develop automated tools for instructional development.

The instructional development process can be divided into a number of different tasks, each of which can be automated to some degree. For example, most instructional projects start with some form of analysis activity (e.g., task analysis, needs assessments). Such activities consist of data collection and interpretation steps. It is possible to use a database program to organize and perform some low-level interpretation of the data collected. If the amount of data collected is large, this can be very worthwhile.

As another example, consider cost/benefits analysis. Most training managers conduct some type of cost/benefits analysis when making decisions about new courses or the use of instructional media (such as CBT or video). The use of spreadsheet programs to evaluate different possibilities can make this form of analysis much easier.

Many curriculum development projects begin with a storyboarding step which has traditionally been done using printed forms. However, powerful graphics editors now make it possible to do storyboards online. In some cases, the information may be used directly in the development of the course materials including print, slides, video or CBT.

At the moment these automated activities are not integrated. However, in the future we will see the availability of full-scale automated instructional development systems covering all tasks. In fact, a prototype of such a system, called the Instructional Toolkit was developed at Courseware as a research project. It used commercially available applications software such as word processing, spreadsheet and database programs. Some of the practical and theoretical issues raised by this project will be discussed.

*Park Row Software
1135-C Garnet Avenue
San Diego, CA 92109
(619) 581-6778

AUDIO-BASED CBT

James M. Kraatz,* University of Illinois

Jolanta Percy, Detroit Edison

Our approach for using computers for teaching or training operating procedures often consists of verbal descriptions and questioning. We try to improve on the situation by introducing visual representations of the procedures, thus bringing the job closer to the computer. Advances in video display technology constantly improve and simplify this task, but producing good simulations is still difficult and expensive. The "RAJA" (Remote Access Job Assistant) system provides a way to bring the computer to the job.

RAJA consists of an Instavox^{TM} random-access audio device, a PLATO or Cluster PLATO terminal, specially modified two-way radio communications equipment and lesson delivery software. The RAJA student "terminal" is a hand-held radio transceiver with an added touch-tone keypad.

The presentation will be a general description of the hardware, a description of the delivery software, and a discussion of instructional design considerations. The system is being implemented for training power plant operations personnel. Some preliminary evaluational data will be discussed. The system will be demonstrated if possible.

*University of Illinois
CERL - 252 ERL
103 South Mathews Avenue
Urbana, IL 61801

(217) 333-6211

MODEL TRAINING PROGRAM FOR RESERVE COMPONENT

A PILOT STUDY

Mahima R. Kundu*

US Army Ordnance Center

The purpose of this presentation is to provide information about the instructional technology used to produce MTP-RC courseware (software) so that other instructional designers can transfer the technology to their own applications. The MTP-RC courseware is highly interactive computer-based training that makes use of high-resolution graphics and randomly accessed videodisc images. Moreover, this training was designed specially for Reserve Component units of the US Army that lack equipment or experienced instructors for training on new sophisticated equipment like M1 Abrams Tank. Also, this training program is very cost-effective as the hands-on training is given by computer-generated simulations and exercises without tying up the actual equipment or instructor manhours. This unique program reduced the training cost to a great extent and this model program can be applied in similar situations where actual equipment (high-cost and high technology) and experienced trainers are not available. The Model Training Program offers unique opportunities for application to other complex weapon systems to meet the training needs where costly equipment and manpower resourcing are not available.

This presentation will describe the results of the evaluation of the training and transfer effectiveness of the MTP-RC courseware on samples of soldiers similar to the target population. The primary purpose of the study (training and transfer evaluation) is to answer two basic questions: 1) Does the MTP-RC courseware train? 2) Does the training received on the CBI system transfer to tasks on actual equipment?

---

*203 Idlewild Street, No. 3D
Bel Air, MD 21014
(301) 278-2736

SIMUEDIT: COURSEWARE AUTHORING TOOL

Dave Lohn-Wiley,* Zandria Merrill

SimuFlite

SimuFlite Training International, a division of the Singer Company, trains corporate, military and commercial pilots. It uses computer based instruction as an integral part of the training process. The original method of constructing computer based courseware proved to be too rigid and awkward to handle redesigns or the subsequent upgrades and changes inherent to the flight training industry. To deal with the problem, SimuFlite has designed its own database authoring tool, SimuEdit. SimuEdit was designed and developed by courseware developers, programmers and instructional designers for their specific needs in maintaining and modifying courseware. It has been expanded over time to also aid in the creation of courseware taking advantage of new directions identified through instructional evaluation and needs analysis techniques.

This paper provides a brief description of SimuFlite and the events leading up to the need to develop an in house authoring tool. A historical synopsis of SimuEdit's development is provided along with an in depth comparison to the old hard code system. Substantial savings are currently being realized in time and manpower. Also, a brief description of some of SimuEdit's general capabilities is given.

*SimuFlite
P.O. Box 619199
D/FW Airport, TX 75261
(214) 456-8000

DESIGNING A COMPUTER-BASED

CRITERION REFERENCED TEST

Francine Manna*

Embedded training courses designed to teach computer applications in a U.S. Army environment were developed. Supplementing each course is a test designed to evaluate student competency, provide immediate feedback and measure training effectiveness.

While it was desirable to recreate the actual operational environment, there are several drawbacks. Therefore, the designer first considered the practicality of testing the actual application. Second, testing time was limited. Third, the number and kind of skills to be tested were examined, resulting in an objective structure consistent with military training procedures.

Test format was determined by limiting the tested skills to those that were the most important, complex and contributed to adequate coverage of the course objectives. In most cases, item format was consistent with the actual application, but slight variations were needed to accomodate the above considerations. Remedial aids were available to provide immediate feedback for incorrect answers.

Four test items were developed for each testable objective to help eliminate the possibility of success by chance and to be consistent with accepted test construction principles. Instant remediation for incorrect answers is consistent with learning theory principles: an explanation of the correct answers, the procedure for doing the item correctly and an opportunity to redo the item is included.

Performance criteria were set at two levels. First, objective mastery was set at getting 75% of the items correct. Second, passing criteria was set at mastering 70% of the objectives. The reporting system provides immediate feedback to the student and allows for the summation of individual results. The summation data forms the basis for identifying shortfalls in the training and flaws in the testing procedure.

*P.O. Box 33368
Ft. Lewis, WA 98433-0368

(206) 967-8598

COURSEWARE DEVELOPMENT FOR THE INSURANCE INDUSTRY

John A. Merrill*

Nationwide Insurance

Nationwide Insurance (headquartered in Columbus, Ohio) currently uses its own Agency Office Automation (AOA) System to enable its agents and sales managers to conduct daily business from virtually anywhere in the country via a store-and-forward computer network linked to the Home Office. In the Spring 1985, Nationwide's Office of Marketing Services created the Division of Computer Assisted Instruction to develop courseware for use on the AOA System. The goals of the division are: 1) to produce high-quality courseware, 2) to decentralize the training function and 3) to reduce conventional training costs.

The creation of the CAI Division is unique within the insurance industry because it is dedicated exclusively to the training needs of the agency sales force. The courseware, which currently encompasses auto, homeowners and life insurance, is used to train novice multi-line agents and to upgrade experienced agents on new products and services. This strategy is designed to increase product knowledge and sales activity while reducing the time spent in conventional training.

A team approach to the courseware production integrates instructional design and courseware authoring expertise with in-house subject matter experts. All courses are thoroughly evaluated and field tested prior to implementation in the Regional Offices. The EDUCATOR by Spectrum Training Corporation was chosen as the courseware authoring tool for its flexibility and potential for broad applications.

*Nationwide Insurance
One Nationwide Plaza
Columbus, Ohio 43216

(614) 249-6242

TRAINING THE MICROCOMPUTER ILLITERATE TO BE PRODUCTIVE USERS OF THE TECHNOLOGY IN A ONE-SEMESTER EXPERIENCE

Richard L. Patterson*

The University of Michigan-Ann Arbor

The purpose is to train a computer-illiterate student to become, in a one-semester course, a reasonably confident, productive user of microcomputer technology, and additionally acquiring an awareness of the potential for creative applications of microcomputers.

An intensive hands-on laboratory approach used by Professor Patterson which, according to student feedback is successful, is organized around a fast moving stream of 25-27 tutorials, each being self contained and 1-3 hours in duration.

Four important actions must be taken by the instructor: selection of an appropriate sequence of topics, provision of text and laboratory materials including software, appropriate supervision and evaluation of progress during lab sessions, and provision of incentives to the student to successfully complete every tutorial.

The laboratory facility available to students includes twenty Zenith 148 micros each equipped with a non-color monitor, dual floppy drives, 640k of RAM and 8087 coprocessor; six Epson Spectrum LX-80 printers; a large white board. Adequate space is provided for each work station which includes a padded swivel chair. Each work station is hard wired into UM Net which is easily accessed through a terminal emulator called WINDOW.

Cost of course materials has thus far not exceeded $35.00 per student, which purchases a loose-leaf text and "studentized" integrated software package. Much other software is provided, one complete set for each work station.

*The School of Natural Resources
The University of Michigan-AnnArbor
Ann Arbor, MI 48109

(313) 763-6961

LEARNER CONTROL: WHERE DO WE GO FROM HERE?

Joellyn Pollock,* Cindy Leshin

Arizona State University

Results of research on achievement effects of self-regulated learning have been inconclusive. While some studies indicate that learner control subjects out perform program control subjects, other studies suggest that learners are not good judges of their enroute comprehension, of the amount of information or practice necessary for best performance, or of the optimal pace the program should progress.

Where do we go from here? First, we need to redefine and limit the term "learner control". Learner control can mean the extent to which an individual believes that his/her actions create an effect. Learner control does not necessitate the learner's retention of total control of the instructional sequence.

Secondly, illusions of learner control without sacrifice of effective instruction are recommended. Program advisement of sequence and scope of lesson based on patterns of responses results in "intelligent" learner control. While minimal program control of pacing is optimal, learner control of pacing when responses are incorrect is preferred. Learner control options should include ability to return to previous information or menus at any time, availability of help and exit procedures and control over order of recording test responses.

Lastly, we need to look at effects of learner control other than achievement. The perception of control over an activity evokes intrinsic interest. How this motivation effects the learner's attention to instructional materials should be considered.

*1628 Palmcroft Drive SE
Phoenix, AZ 85007

(602) 257-4327

PROBLEMS AND STRATEGIES FOR DEVELOPING CBI TO MIRROR A MAINFRAME SYSTEM

Mary Riley, Shelley L. Morse*

University of Maryland

This presentation will discuss a co-development effort between the Center for Instructional Development at the University of Maryland, University College (UMUC/CIDE) and a major production company to design a comprehensive training program utilizing print-based, videodisc, computer-based and live simulation.

The recipients of this training are newly hired claims adjustors at a leading national health insurance company. The project's goal was to redesign an existing twelve week traditional training program into an interactive six week program.

Part of the training problem was that current training provided minimal hands on computer experience to new claims adjusters whose jobs entailed a major percentage of time entering data on a mainframe system.

UMUC/CIDE's charter was to design a computer-based training (CBT) component that would be used in conjunction with new lecture/workshop materials. The design strategy was to develop tutorials which provide drill and practice with feedback for selected aspects of the actual system.

This presentation will discuss design strategies/issues in: 1. developing customized courseware using a fairly rigid authoring system (PLATO); 2. translating courseware from a mainframe development environment to a microcomputer delivery environment; and 3. using the PLATO development system and IBM PC delivery system to display screens and emulate behaviors indigenous to the mainframe system the trainees would eventually be working with.

*University College CIDE
University of Maryland
College Park, MD 20742

(301) 985-7370

CORPORATE-WIDE COMPUTER MANAGED INSTRUCTION (CMI)

Joan Rogalska Saz,* Applied Science Associates, Inc.

David F. Stevens, AT&T

AT&T has developed a set of corporate-wide functional requirements for Computer Managed Instruction (CMI). This effort is at the heart of a common operating environment for electronically-delivered training within AT&T. With a generally understood definition of CMI requirements, implementors of the environment can provide the capabilities needed for course assignment, testing and record-keeping. Similarly, CBT developers can design courses with the assurance that their CMI features will work as intended.

This session describes the methodology used to define CMI standards corporate-wide for the first time, bringing together disparate groups of CBT users to develop a common view of the CBT delivery environment. The presenters will discuss the interview procedures used to obtain requirements of individual organizations from experienced members of the CBT community throughout AT&T. The presentation will also describe how the needs of these organizations were synthesized into a set of general requirements and the structured walk-through process by which organizations agreed on a final set of standards. Participants will also learn how general CMI features can be customized to meet the needs of individual groups within the company.

*Applied Science Associates, Inc.
292 Three Degree Road
Butler, PA 16001
(412) 284-7374

COMPUTER BASED TRAINING (CBT) COST ESTIMATING ALGORITHM FOR COURSEWARE (CEAC)

Ralph E. Schooley*

Space Applications Corporation

A Computer Based Training (CBT) Cost Estimating Algorithm for Courseware (CEAC) development is communicated. The algorithm enables a user to assign values to parameters that define the instructional design strategy, the pedagogical base, the experience of the courseware developer(s), the availability of production-oriented libraries (e.g. utility, application, graphic), the adequacy and availability of productivity tools and cost primitives (e.g. developer salaries, burdened labor rate, general and administrative (G&A) rate, and length of work month). Upon capturing the assigned parameter values, the algorithm employs tabularized engineering judgement values to estimate development costs. The data retrieval and computational requirements of the algorithm can be easily implemented on an electronic spreadsheet. A specific implementation is examined using practical decision-making scenarios pertinent to various aspects of CBT project design and productivity.

Use of the CBT CEAC approach during the bid and proposal stages of contract negotiation can provide management with a technical basis of cost estimate, as well as with a valuable risk assessment tool. Use of the CBT CEAC approach by prospective CBT users provides insights into the feasibility of implementing CBT within existing budget constraints.

*19900-A Olivewood Street
Cupertino, CA 95014
(408) 996-8737

CORPORATE CHALLENGE

COMPUTER-BASED INSTRUCTIONAL GAMING

Linda Shatzer*

AT&T

As part of meeting the need to provide orientation to new employees, computer-based training was selected as the most appropriate delivery system. A goal of this project was to prepare the new hire for the work environment by active participation in a decision making process. As a result, instructional gaming was selected as the design technique for the project. The name of the game, which reflects the theme of the course, is "Corporate Challenge". In this way, information about the company could be presented to the employee. In addition, through the choices offered to take "risks" or select "opportunities", questions about the content could check for understanding while motivating the student.

The method to design the content of each module, ideas for creation of display screens and the programming were developed to meet the specific needs of the target audience.

This presentation will describe the content of each module, explain the instructional game, illustrate graphics used on a personal computer, demonstrate testing and feedback and provide an evaluation of the project.

Instructional gaming which might be looked upon as a less than serious way to train new employees can be a powerful and motivating tool for orientation to a new company.

*AT&T
399 Campus Drive, Room 13B20
Somerset, NJ 08873

(201) 271-7460

CHECKING VENDOR SUPPORT

Timothy W. Spannaus*

Tim Spannaus and Associates

Companies that purchase a CBT package want to build an organized, well-rounded training program that will assure them of a steady supply of sound courseware and consistent improvements in trainee performance.

Although the actual selection of the right CBT package is vital, people often fail to pay attention to an item that is crucial to proper utilization of the package: vendor support.

There are two main issues in identifying vendor support, related to the product itself and to the CBT business in general.

Product support issues are those related to a specific CBT system. Product support can include documentation, training, newsletters, electronic bulletin boards and more. There are pricing implications to product support that you should consider.

Beyond a specific CBT product, you may want to be reassured that the vendor intends to stay in the CBT business. Such a commitment would imply that the vendor would continue to develop new products and services to help you in your CBT efforts. Consider financial stability and the importance of CBT to the vendor.

The importance of support varies with the kind of project you are undertaking. Some projects require a well-supported system; others will tolerate poor support.

---

*Tim Spannaus and Associates
22933 Coventry Woods Lane
Southfield, MI 48034

(313) 357-0004

LEARNING STYLES, COMPUTER ATTITUDES AND PREFERENCES FOR INSTRUCTIONAL DELIVERY METHODS

Charles J. Stanley,* John C. LeDoux

FBI Academy

The presenters will discuss the preliminary results of an ongoing study of possible relationships between individual learning styles, attitudes toward computers and learner preferences for instructional delivery methods. Adult learners enrolled in a National Academy introductory microcomputer class served as subjects in this phase of the study. The class incorporates both instructor-led activities and computer-based training while stressing hands-on learner experiences. At the end of the course each student was administered Kolb's Learning Style Inventory (ILS), the Computer Attitude Scale (Loyd & Gressard, 1984), and an instructional methods evaluation form.

Some interesting findings were noted. Seventy seven percent of the students indicated that the computer-based training was more helpful than the instructor-led activities in learning the subject matter. The same percentage of students indicated that they "liked" CBT better. Individual learning styles alone did not seem to differentiate student preferences for one instructional method over another. However, expressed attitudes towards computers did. Surprisingly, a far greater proportion of the students with less positive attitudes towards computers tended to prefer CBT over the instructor-led activities than did those with more positive attitudes.

*FBI Academy
Quantico, VA 22135
(703) 640-1192

AN AUTHORING SYSTEM SELECTION GUIDE

David F. Stevens,* AT&T

Joan Rogalska Saz, Marjorie Pomper

Applied Science Associates, Inc.

AT&T has produced an authoring system selection guide. The guide, to be revised annually, helps training managers inexperienced in CBT development choose an authoring system to meet their current and future needs. With the guide, users can discriminate among eight "qualified" mainstream authoring systems (not identified in the presentation) on the basis of the production and delivery requirements of their own organizations.

The presenters will discuss the features that served to qualify an authoring system as possessing "necessary and sufficient" instructional capabilities. The presentation will also describe a set of structured questions that guide a manager in defining the production and delivery environment in which the authoring system will be used. The manager's answers to the questions narrow the field of candidate authoring systems to one or two. Where two candidate systems exist, a brief table guides the user to the consideration of additional delivery and production issues and to a choice based on weighted ratings. Participants will learn: 1) How the "necessary and sufficient" instructional capabilities were identified, and how authoring systems were determined to be "qualified". 2) How a set of structured questions on delivery and production requirements was identified.

*AT&T Corporate Training Support Group
219 Mt. Airy Road
Basking Ridge, NJ 07920
(201) 953-6168

INTERACTIVE VIDEO ON A SHOESTRING -- REVISITED

L. Boyer*, L. West

North Shore Community College

North Shore Community College, a medium-sized community college in coastal Massachusetts, is committed to instructional innovation via media use. Unfortunately, our funds do not always match our ideals. Nevertheless, when the coordinator of the tutoring program for the Academic Assistance department proposed designing and producing an interactive video program, the administration was all for it. This presentation details the interactive video development process from initial concept through production.

In May 1986, Lois West proposed producing an interactive video program to simulate the peer tutoring process. We formed a small project team (including the Coordinators of Educational Technology and Instructional Media Services), and developed the idea into a two-part instructional simulation comprised of an initial interview session as a composition tutor and a follow-up interview session as a mathematics tutor. The initial project plan was approved.

In the Fall of 1987, after procuring a course release for Ms. West, we began project development work. We used a simplified instructional systems design model (needs assessment, analysis, design, development, evaluation, revision) to guide our process. Working from our task analysis flowcharts, we produced the script -- modified free-branching simulation -- and developed storyboards. At this time, we are in the midst of shooting the video component during Spring semester 1987. The college has provided release time for both coordinators to complete the video and CAI components of the project. Using the BCD system and SuperPILOT, we will produce a 3/4" master tape by May 30, and implement it in training sessions with students in Fall 1987.

*1 Highland Avenue
Nahant, MA 01908
(617) 593-6332, Ext. 280

A FEATURES COMPARISON OF PRINT SHOP FOR THE APPLEII, MACINTOSH AND IBM

B. Clark, W. Clark - Gardnerville Elementary School

N. Higgins* - Arizona State University

Application features of the graphics program PRINT SHOP will be compared for three different computers. The presentation will focus on the ease of learning for both beginning and experienced computer users and on the different levels of complexity that each type of machine offers. Output devices, including the use of laser printers will also be discussed. Visual displays will illustrate the differences in quality.

PRINT SHOP is popular in school settings because of its great versatility. It is used for preparing graphic materials, artwork for overhead transparencies, slides and video reproduction, and artwork for "paste-up" on printed materials. PRINT SHOP also has many corporate uses including making charts, graphs, titles, bullet slides and other speaker support transparencies. Signs and directions for point of use on demand training are easily produced with this software. Samples of possible school projects and examples of corporate uses of the program will be shown.

*Educational Media and Computers
College of Education
Arizona State University
Tempe, AZ 85287

(602) 965-3651

EFFECTIVENESS OF WRITING TO READ PROGRAM IN PHOENIX PUBLIC SCHOOLS IN ARIZONA

Patricia Alba Mulligan*

Arizona State University

The IBM Writing to Read Program, developed by Dr. John Henry Martin, is a computer-based instructional system designed to develop the writing and reading skills of primary school children. The system works within the context of a planned learning center, a room where students use a variety of equipment and materials organized at four learning stations. At the stations the children use a computer with a set of instructional and game diskettes, a cassette tape recorder and various sensory materials.

The Phoenix Elementary School District in Phoenix, Arizona purchased the program for its thirteen schools. A study is being designed to determine if the WRP increases reading test scores on the Arizona State Criterion Referenced Reading Test (CIDS). The design is Group Within Treatments and test scores will be analyzed using Multivariate Analysis of Variance procedures (MANOVA).

Treatment and control groups have been identified in each school. MANOVA will test for gender, ethnicity and grade as well as treatment effects. The dependent measures will be the subtest scales of the CIDS Test and will be identified as post treatment administrations.

Results will be presented at the Conference.

*2563 North 22nd Drive
Phoenix, AZ 85009

(602) 253-5811

COMPUTER-BASED TESTING IN SCHOOLS

James R. Okey,* Chris L. Peters, William H. Brittain

University of Georgia

Computers as tools for both teachers and students have played important roles in school computing. Computer-based testing is one of the important and growing tools in classrooms. Numerous programs exist to aid instructors in generating, administering and scoring tests. During a time when increased calls for school testing are being made the computer holds considerable promise in accomplishing testing goals without unnecessarily increasing time investments.

The purpose of this presentation is to examine some of the roles that computer testing can play in classrooms. Consideration will be given to what kinds of testing computers are best suited and areas in which they have limitations. Special emphasis will be given to the effect of computers on the quality and efficiency of classroom testing. A number of resources for classroom testing using computers will be examined and evaluated.

The final purpose of the session is to share experiences (both successes and failures) in implementing computer-based testing in school systems. A testing system designed for stand-alone and networked use will be described. Problems in adapting this and other systems to school use will be discussed.

*Department of Instructional Technology
University of Georgia
Athens, GA 30602
(404) 542-1763

HIGH TECH'S REAL PLACE - WHAT WILL IT TAKE?

Dan Schwartz*

Continuous Progress Learning Association, SALT

High Tech only compounds the problems of the self-contained lock-step classrooms where teachers are assigned 30 students of the same age. The teacher's major effort is to keep the class moving together at some level relevant to the entire class. Aside from helping slow learners and providing activities for the others, High Tech increases the performance spread further compounding the teacher's problems. Unless the methods of assigning students to classes is changed to a performance based rather than an age based system, with each teacher handling different performance levels, High Tech is going to be junked along with all the other teaching machine methods used for delivering instruction.

The answer lies in changing the variable from amount learned to time necessary for mastery. Students in each class have mastered the lower levels, therefore, the subject matter is easily learned. The mood is to achieve. Discipline problems are greatly reduced. Teachers can produce software, stock software and collect materials for a limited number of levels. With High Tech delivering the majority of the instruction the teacher has the opportunity to take small groups for in-depth discussicns, thinking skill practice and scores of other activities. There is time to help slow learners.

*4942 College Avenue
Riverside, CA 92505
(714) 687-8617

IMPLEMENTING AN APPLICATIONS PROGRAM DATABASE

Scott D. Shoemaker*, Norman C. Higgins

Arizona State University

Educators find that their efficiency and quality of work is enhanced by using writing, authoring, graphics or information management software. Locating and selecting quality applications software, given hardware and cost constraints can be time consuming. Access to a relational database containing information on applications programs such as cost, publisher, machine compatibility and references to reviews would simplify the selection process.

This presentation will describe the process of designing and implementing an applications programs database. Problems and solutions to creating the database using dBase III plus and tryouts of instructions for using and updating the database will be discussed. How the database is used to individualize instruction in a computer applications class at ASU will be described.

The presentation will include a demonstration of the use of the database, allowing the audience to interact with it. A data disk will be made available for individuals to copy.

*1014 E. Spence, #104
Tempe, AZ 85281
(602) 921-0517

VALIDATION OF A COMPUTER ADMINISTERED RHYTHM TAPPING SKILL TEST FOR ELEMENTARY SCHOOL STUDENTS

Aaron L. Spaulding, Gerald J. Blumenfeld*, Ted Malcolm

The University of Akron

An evaluation of programs to teach rhythm skills to elementary school students was undertaken. Reeves and Lent have provided some guidelines for the evaluation of courseware. Within their scheme, a major emphasis is placed upon the need for reliable and valid performance measures of the learning outcomes promised by an instructional program. The primary emphasis of this study was to develop such an instrument.

Music Theory, Vol. 1, and Mr. Metro Gnome were the two programs to be evaluated. Both endeavor to teach rhythm tapping skills. A portion of the MECC program was modified so that it could be used independently as a test. Fourth, fifth and sixth grade children were used as subjects to collect data on the reliability and validity of the test. Data was also collected on the instructional effectiveness of the two programs. A reliability estimate was found to be .93, while the validity estimate obtained was .63. Both programs appeared to be very effective in improving the rhythm tapping skills of the elementary students.

*Department of Education Foundations
The University of Akron
Akron, OH 44325
(216) 375-6956

The Effectiveness of Using Computers in Developmental Writing Classes

Gary Vaughn*

University of Cincinnati

During the 1986-87 academic year, an ongoing study was begun at the University College of the University of Cincinnati to measure the effects of classroom computer use on the composing processes of developmental writing students. The study involved five English instructors and eleven sections of a developmental writing class with a combined enrollment of 150 students. The students were broken into three groups: Classes that used the Bank Street Writer word processing program on Apple IIe personal computers; classes that used both Bank Street Writer and Writer's Helper, a guide to the writing process; and control classes that did not have computer access. All classes followed a common syllabus and used a point-based grading system. Students had been placed into the developmental classes on the basis of their scores on the Development of Reading Powers test and on a holistically graded essay. At the end of each quarter, the developmental students retook the essay test, a passing grade being required for placement into Freshman English. Evaluations used in the study included class grades, test scores and questionnaires. This presentation will explain the methodology of the study and will explore the first year results and their implications.

*CAI Laboratory, ML #203
University of Cincinnati
Cincinnati, OH 45221

(513) 475-6259

ELECTRONIC SUPPORT NETWORKS FOR IMPROVING PRESERVICE AND INSERVICE TEACHER TRAINING

Michael Waugh*, James A. Levin

University of Illinois at Urbana-Champaign

We are implementing the notion of "electronic support networks", which interconnect students, teachers and university faculty. The purpose of this networking activity is to foster collaboration among teachers and learners who are geographically dispersed. The topical nature of the interactions conducted via the network so far have included joint problem solving activities in science, mathematics, language arts, microcomputer applications and software evaluation.

The interaction of these groups of individuals through electronic communication networks can improve university-level preparation of teacher candidates, while at the same time improve the knowledge and skills of practicing teachers. In addition, participation in these instructional networks by teacher candidates will give them the skills and knowledge necessary to continue their education using this new instructional medium following graduation.

Our concept has been to "institutionally ground" these network interactions. At the university level, the network support provided to schools is justified by its instructional value, since question answering (communication of knowledge and understanding) is part of existing university courses. It has the secondary benefit of providing university students with computer literacy experience and providing for the development of computer skills among university faculty. For the precollege schools participating in the network, it is justified because of its contribution to the content of classes currently being taught to those students. It has the secondary benefit of contributing to inservice faculty training both within specific content domains and with computer skills.

At both the college and precollege level, it enriches the instructional environment, increasing student/teacher contact and the development of communications skills. It opens up both the college and precollege settings to a wider range of resources "in the real world". And, it provides a way for universities to make concrete contributions to schools and to gain from this involvement in their real world questions and problems. It also provides a way for precollege schools to access wider expertise, and for them to assist in teacher preparation so that new teachers develop more realistic expectations and acquire more relevant skills and knowledge.

*386 Education Building
University of Illinois
1310 South Sixth Street
Champaign, IL 61820

(217) 333-0227

EFFECTIVE USE OF PLATO IN THE K-12 SCHOOL SYSTEM

Tom Williams*, Marieanne Perrault, Mike Pineda, Lee Becea

Sweetwater Union High School District

Faced with an increasing dropout rate, the Sweetwater Union High School District opened its first PLATO Independent Learning Center in April 1986. Shortly thereafter, five more centers were opened utilizing the PLATO/CSUS (Cal State University Sacramento) network. Over 1,500 dropout or high-risk students have been served in a uniquely structured educational program. PLATO, directed study and independent study have been incorporated to give the visual/kinesthetic learners another opportunity in an innovative educational setting.

PLATO is an acronym for "Programmed Logic for Automatic Teaching Operations". PLATO is the world's largest collection of computer-based education courseware and over one billion dollars have been invested in the development of this very sophisticated educational library. It is the only objective-based continuum curriculum that can be effectively integrated into the California Framework of Curriculum.

Pre and post testing on the Test of Adult Basic Education indicate a gain of 1.8 years in reading and 2.2 years in math over a nine week period. Students tested also relate a renewed desire to continue their education. PLATO Centers offer all course requirements for a regular high school diploma.

Dozens of students have graduated as a result of passing proficiency tests and earning credits in the PLATO Centers. Fifteen percent (15%) of the students served are transferred back into the regular high school program with renewed self-confidence and self-esteem. The many benefits of computer-based education are being explored and implemented in the Sweetwater Union High School District.

*5504 Horseridge Way
Bonita, CA 92002
(619) 691-5430

COMPUTER BASED EDUCATION
IN CALIFORNIA PUBLIC SCHOOLS

Pamela A. Wright*

California State University

California State University (CSU) provides central site PLATO to California Public Schools and has been experiencing highly successful results. The CSU network has the capacity to provide CBE to the Western United States. The current application in public schools is aimed at the truant/dropout population and as a result the program generates its own revenue. The courseware provides a complete continuum of curriculum which integrates appropriately with the California Framework of Curriculum. The courseware is both mastery based and objective based which contributes to the rapid achievement experienced by students. The management system provides for complete track and flow of the student and creates truly individualized instruction.

Recognizing the financial constraints in school districts, computer services have been designed to meet a plethora of educational challenges along with the strategy to generate revenue to fund a PLATO learning center. This concept makes it possible to generate otherwise lost revenue and thus cover the cost of education computer services. The aim is to focus on the truant/dropout population and bring them back into the educational system. Both research and experience have clearly established that PLATO is a highly successful means of retrieving this population which is characterized as being the visual/active learner; the learner who interacts best with technology, but will not perform well in the classroom which is designed for the auditory learner.

In addition to this basis of revenue, other programs can be considered to support computer services. Therefore, it is conceivable that mainframe PLATO can serve a multiplicity of populations with varying needs and this workshop is designed to share the successful models.

*27864 Emerald
Mission Viejo, CA 92691

(714) 364-1351

EDUCATING SPECIAL LEARNERS: A COLLEGE COURSE

G. Phillip Cartwright*

The Pennsylvania State University

Educating Special Learners is a stand-alone computer assisted college course. It consists of 18 double-sided disks built around 18 content modules, a practice quiz disk for each module and a set of instructor's quiz disks with 400 items. Approximately 35 to 40 clock hours are required for completion of this course.

To satisfy college and university administrators at Penn State and elsewhere, three criteria were used to evaluate the full-length course: academic soundness, attractiveness to students and cost effectiveness. During a 2-year evaluation, Penn State campuses as well as colleges and universities outside of Pennsylvania evaluated the course on three criteria. In addition to these data, information is available about developmental and implementation costs and research programs accompanying this operational course.

*Division of Special Education and
Communication Disorders
124 Moore Building
The Pennsylvania State University
University Park, PA 16802
(814) 865-6072

COMPUTER PROJECTS AND PRODUCTS FOR HEARING IMPAIRED POPULATIONS

Phil Mackall*

Gallaudet University

Educational settings for hearing impaired students are scattered widely around the nation in special schools and in mainstreamed programs. Communication among those programs about new hardware and software development is often limited and not timely.

This presentation will share three electronic mail communication links that can help to close the information deficit. In addition, new products such as a public domain shopping service, a national software evaluation and lending library, a new TTY/modem/answering machine board for IBM's and more will be described. A handout will be distributed listing many of the current activities with computers at the elementary and secondary level, including project title, project administrator, contact information and a brief project description. Highlights of some of the most exciting will be related.

*Pre-College Programs
Gallaudet University
800 Florida Avenue NE
Washington, DC 20002

(202) 651-5043

USING COOPERATIVE LEARNING TEAMS IN THE RESOURCE SPECIALIST PROGRAM WITH COMPUTERS

Eleanor G. Thompson*

This study was devised by a Resource Specialist team using learning disabled students in the regular education elementary classes. The RSP team attended district supported cooperative learning seminars and planned lessons using cooperative learning strategies combined with specific computer software. The student teams were four to six members of mixed-abilities with various handicaps.

The study had three questions: 1) Can students learn and remember social studies and English units better when aided by computers? Specific handicaps: memory, organization, sequencing events, dysgraphia, dyslexia. 2) Does student learning improve in the area of the curriculum addressed when cooperative learning teams are using and developing computer lessons? 3) Will student attitude concerning leadership, cooperation, pride in team work and responsibility improve? The W-J Battery was given to all students before beginning the study and post scores were compared. Team analysis sheets for each student were provided before and after a lesson unit to determine an increase or decrease in attitude.

The RSP classroom had a computer and a computer laboratory consisting of 21 microcomputers was available. Class size was limited to 12 RSP students per period. The student teams were introduced to cooperative learning methods and video taped before and after each unit. Results showed increase in confidence and clarity in speaking about the goals and objectives of their cooperative learning teams. Students then returned to their regular classrooms and presented the lessons using a computer projector and video tape presentations. In addition, students gave demonstrations for other Resource Specialists. RSP students showed an increase of 1 to 1.5 (average) on the W-J PsychoEd Battery in the Reading and Written Language clusters. Weekly surveys indicated that all students gained a more positive attitude toward school and especially toward the Resource Specialist program. In addition to the academic success, students gained self-esteem by being the "experts" in their regular classes.

*101 Peter Coutts Circle
Stanford, CA 94305
(415) 494-3303

LEARNING BEHAVIORS OF BACCALAUREATE NURSING STUDENTS AND COMPUTER-BASED EDUCATION

Jean Arnold,* Gail Iglesias

Rutgers The State University

The purpose of this research is to evaluate the use of computer-based education with sophomore nursing students. The literature related to computer-based instruction and problem solving indicates that learning behaviors are influenced by several variables. Thus three measures toward computers were administered to a sample of fifty sophomore nursing students, prior to the introduction of tutorial computer-assisted instruction. Tutorial software includes several skills, concepts, rule application and problem solving (Bell, 1985).

Attitudes toward CBE were measured using a semantic differential tool (Allen, 1986), Computer Attitude Scale and a Problem-Solving Attitude Scale (White, 1986). The reliability of these instruments ranges from .82 to .85. One group was given tutorial CAI related to the problem solving achievement measure and the other was given tutorial CAI related to basic nursing skills.

The presentation will include data analysis which is incomplete at this time, since the research study period is Spring Semester 1987.

*College of Nursing
Rutgers The State University
180 University Avenue
Newark, NJ 07102
(201) 648-5082

EVALUATING INTERACTIVE TRAINING FOR ACUTE TRAUMA LIFE SUPPORT

Mary R. Marlino,* Thomas C. Reeves - The University of Georgia

Joseph V. Henderson

The Uniformed Services University of the Health Sciences

The Advanced Combat Trauma Life Support (ACTLS) Interactive Videodisc Case Simulation was designed to meet the needs of training military physicians in battlefield trauma management in a peacetime environment. The evaluation of this program consisted of a beta test and a formative assessment corresponding to the first two levels of evaluation recommended for interactive instructional programs by Reeves and Lent (1984), program documentation and formative evaluation.

The evaluation was conducted at the Medical Training Facility at Fort Sam Houston in San Antonio, Texas, over a twelve week period. During this time, a sample of seventy-two physicians used the program. A variety of evaluation designs and methods, including equipment logs, anecdotal records, questionnaires, physician interviews and computer-collected data were utilized. The evaluation addressed issues such as the functional reliability of the ACTLS hardware and software, the content validity of the ACTLS simulation, user interface and documentation enhancements, physician and instructor reactions, and overall program enhancements.

The Advanced Combat Trauma Life Support IVD Simulations delivery systems consists of an IBM PC microcomputer, a Pioneer laser videodisc player, and a 12" Sony monitor equipped with a light pen. All patient evaluation, testing, diagnosis and treatment decisions are made via lightpen or keyboard access from a series of "windowed" menus generated by the computer. The computer program is written in the C programming language.

*Instructional Technology
The University of Georgia
607 Aderhold Hall
Athens, GA 30602

(404) 542-3810

THE HURTING TOOTH

AN EXPERT SYSTEM TO DIAGNOSE TOOTH PAIN

Monica Nagy,* Ernst Schelb

University of Texas-San Antonio

A patient arriving at the dental office may complain that a particular tooth hurts, but in reality, the cause of the pain can be another tooth or some other part of the mouth. This program does not deal with determining which tooth, if any, is causing the pain. Instead, it is to be used after the suspect tooth is found. It diagnoses the vitality of the pulp tissue in the suspect tooth that is causing the pain.

By working through the expert system, the user learns the proper method of diagnosing tooth pain: what to ask, when to ask it and what to conclude from the information. By varying their responses, the users can practice different situations. The rules for achieving the diagnosis can be displayed at any time for review of the process. In case of disagreement, each rule contains a suggested reference. The expert system also can be used as a friendly second opinion in cases of doubt.

The expert system was developed to be used by freshmen through junior dental students to practice real life diagnoses of tooth pain problems. This is important because the average dental student has very few opportunities to practice differential diagnosis of tooth pain. Endodontics is an area of dentistry which lends itself to an expert system because the criteria for diagnosis of tooth pain is well defined.

The presenters will describe the design of the early prototypes and will demonstrate these early versions of the system written in different languages. The benefits and limitation of each language also will be discussed. Finally, the expert system in its current form will be demonstrated.

*Division of Instructional Development
University of Texas Health Sciences Center
at San Antonio
7703 Floyd Curl Drive
San Antonio, TX 78284

(512) 567-2280

HELP: DEVELOPMENT OF A HOME ECONOMICS MICROCOMPUTER DATA BASE SYSTEM

I. Hathaway,* M. L. McPherson - Michigan State University

R. Heatley - Cooperative Extension Service, Jackson County

Home economics content is continually changing. As research, changing conditions and new products affect the lives of families, an Extension home economist's role as an information source evolves. Staying ahead of the information flow, transferring and interpreting information can be solved with the use of computer technology.

The Michigan Extension Home Economics program has updated our information delivery system by developing Home Economics Library Programs, HELP, in two subjects: food preservation and home maintenance and repair. HELP will be used to 1) answer clientele questions, 2) produce materials for mass media through an interface with word processing packages and 3) give demonstratios with a laser disc at fairs and malls.

Our goal was to develop a standardized system that would allow for one common search and retrieval method as well as one interface with MultiMate. HELP has been developed using dBMan which has the ability to use a key word system with extensive cross referencing within an existing software shell.

HELP has applications for all areas of home economics and can be used as a model for effective data base development.

*Department of Family and Child Ecology
Michigan State University
203 Human Ecology
East Lansing, MI 48824
(517) 353-3886

COMPUTER APPLICATIONS IN FAMILY RESOURCE MANAGEMENT

Arlene Holyoak*

Oregon State University

Many institutions of higher learning have not incorporated micro computer applications into specific content curriculums. Various concerns and challenges in using computers in resident instruction include: lack of hardware, support systems and instructor knowledge, plus software concerns such as availability of quality programs, copyright agreements, cost, etc.

Although many challenges do exist, advantages and benefits accrue from using computers in resident teaching. Some benefits include: 1) increased computer skills enable students to compete more effectively in the computer age job market, 2) generating interest and excitement in learning, 3) providing the advantage of individualized instruction, 4) providing opportunities for students to simulate experiences that would be difficult or impossible in everyday life and 5) increasing credit hour production.

This presentation will share various ways in which computers are being used in resident instruction at the university level in family resource management. At Oregon State University, financial management applications include the use of commercial software packages and spread sheet applications and computer-assisted design and drafting (CADD) in housing. A statistical package also teaches basics of the research process to undergraduates and is used in graduate student research. A final use is in classroom administration, including computer generated examinations and record keeping.

*Family Resource Management
Oregon State University
Corvallis, OR 97331

(503) 754-4992

SEEK AND FIND IT WITH A DATA BASE

Dorothy Jensen*

University of Wisconson - Stout

Most everyone has experienced the frustration of finding some item that they are sure is somewhere, but where.

This can be the same frustration when trying to find an item in a large collection of historic garments, accessories, lace samples and other textiles owned by the Apparel, Textile, Design Department at University of Wisconsin-Stout.

There is a need to know exactly what items are in the collection for insurance and inventory, provide a convenient list for teachers and students to find items for display or study purposes. Fortunately, the right factors needed such as some available funds, a person who understood computers, as well as someone to push the idea to catalog the collection by computer and a database program were available.

The entire project to catalog all items was accomplished in 1 1/2 years and at a one class credit assignment. PC-FILE was the database program used and to accommodate the items it was necessary to use a separate disk for 1) clothing items, 2) accessories, 3) lace collection and 4) other textiles.

The department also has a large collection of dolls dressed in historic garments which has yet to be added to the database. We also need to continue to add the newly obtained items.

*110 Hillcrest Road
Menomonie, WI 54751

HOME ECONOMICS NETWORK:

PROGRESS AND PROMISE

Ann B. Kapp*

Northeast Louisiana University

Improvement of teaching of all subject matter areas is a top priority of educators everywhere. In Louisiana, one innovative way to achieve this in the home economics classroom has been the introduction of a telecommunication network where home economics teachers can access information and teaching ideas for the different subject matter areas of home economics. They can also share effective teaching ideas and methods that they have discovered in their own classrooms.

An in-state WATS line connected to a Tandy 1000 HD computer using the Bread Board System software located in the Department of Home Economics at Northeast Louisiana University is the source of information for the teachers. The system is monitored and maintained by faculty members of the department. Teachers can request specific help from university professors.

Successes and failures of the program will be discussed. People problems like the difficulty of convincing secondary teachers to get involved in the new technology and methods for convincing them to try will be included. Technical problems including those concerned with configuring a bulletin board program to meet the unique needs of the classroom teacher will also be discussed.

*Northeast Louisiana University
Department of Home Economics
Monroe, LA 71209-0560

(318) 342-2155

COMPUTERS IN HOME ECONOMICS:

GRADUATE LEVEL TEACHER EDUCATION

Ann S. McChesney*, Cook College - Rutgers University

Lynne Levy, Glassboro State College

Elaine Muller, Computer Education Consultant

Recognizing that many home economics teachers lack the knowledge and skills necessary to incorporate computer applications into their instructional or departmental management functions, the state education agency funded three geographically distributed graduate courses for computer inexperienced teachers. Sixty learners (twenty per site) were provided "hands-on experience", during the 1987 "Computers in Home Economics" course. Objectives, course outline and evaluation criteria were similar at each site. Objectives encompassed creating interest, promoting attitude change, increasing knowledge about hard and software selection, operation and use, and focused upon both management and instructional applications. Analysis of student's individual school resources was included; learners were encouraged to borrow software and experiment with their students.

Pre and post data relative to: attitude concerning computers in home economics, computer skills, applications (personal and classroom), availability of funds, and anticipated actions were collected. An additional follow data collection in Fall 1987 will focus upon actual changes in effect. Data analyses relative to these variables and student demographic data as: content areas taught, prior computer experience and type of school will be completed.

*Department of Home Economics
Cook College, Rutgers University
New Brunswick, NJ 08903

(201) 932-9516

INNOVATIONS IN COMPUTER AIDS:

THE DEVELOPMENT OF A QUICK-REFERENCE MANUAL AND A SLIDE TRAINING PROGRAM

Phyllis Bell Miller*, Jacquelyn DeJonge

College of Human Ecology - University of Tennessee

In response to the need for clearer, more concise computer aids, we have developed a unique computer reference as well as instructional slide sets. The reference covers the general use of the software, while the slides also include apparel and interior design applications. Although these aids were written for CAD software (AutoCAD (TM)), the same concept can be applied to any software.

The reference includes brief definitions of each command and a step-by-step guide, showing screen prompts, appropriate responses, what should occur after each response and examples. It is also engineered for convenience. In experimental comparisons between subjects with similar levels of computer experience, subjects using the new reference were able to execute commands an average of 250 percent faster than those using the manual.

Providing a new approach to instruction, the slides show the steps of selected commands or applications. During demonstrations, students remain at their terminals and work at the same pace. Steps can be easily reviewed. The slides are also ideal for individual training and review.

*Department of Textiles, Merchandising and Design
College of Human Ecology
The University of Tennessee
Knoxville, TN 37996-1900

(615) 974-2141

APPLICATION OF AN ELECTRONIC SPREADSHEET TO A FOODS/FOODSERVICE CURRICULUM

Barbara F. Quinby*, John Cheyer

Manchester High School

Home Economics teachers are being confronted with the task of integrating "basics" within the curriculum. This presentation will explain an innovative method of using electronic spreadsheets to understand mathematical computations used in meal planning and preparation. Once students are introduced to a teacher prepared meal via the spreadsheet using templates, they are able to understand the "mechanics" of how it works. Their final project consists of designing and presenting a meal or product to the class.

*Manchester High School
134 E. Middle Turnpike
Manchester, CT 06040

ESSENTIAL ARITHMETIC SKILLS

VIA INTERACTIVE VIDEO

Edward C. Beardslee*

Millersville University of Pennsylvania

Dr. Edward C. Beardslee, Associate Professor of Mathematics and Computer Science at Millersville University of Pennsylvania supervised a team of fifteen mathematics subject-matter experts to author the content for an interactive-video based course in essential arithmetic skills for adult learners. The target audience includes students enrolled in remedial arithmetic courses at junior and four-year colleges, technical and business schools, GED preparation and minimum competencies for high school graduation.

The course is being developed by ISC Educational Systems, a Lancaster, Pennsylvania based company specializing in developing interactive-video education and training material. Before ISC releases the course, an extensive field test will be conducted at Millersville University of Pennsylvania. Dr. Beardslee and his staff will coordinate the field test and course revision.

The arithmetic skills course is taking two years to develop. Included was a detailed three-month program planning phase, a two-month design phase and a three-month analysis phase. The content writing phase lasted three months. Approximately six months was allotted for video-production and programming. Finally, a six-month testing and revision phase completed the course development.

*Department of Math and Computer Science
Millersville University of Pennsylvania
Millersville, PA 17551

(717) 872-3531

INTERACTIVE VIDEODISC PROGRAM TO
LEARN TO RECOGNIZE JAPANESE

Alfred Bork*

University of California

The Educational Technology Center at the University of California, Irvine, is joining with groups in the United States and Japan to develop a course in Japanese, for English speakers. The course will use both computers and videodiscs, to promote highly interactive learning and environments for a wide audience. Currently the project is in pilot phase. Ambassador David Osborn, now in San Diego conceived the project. The support currently is from Fujitsu Ltd. and Nippon Television Cultural Society. Linguists and teachers of Japanese from many universities are cooperating in the project. The emphasis in this initial phase is on teaching students to recognize spoken Japanese rather than teaching them to speak or read the language.

Students will watch portions of various NTV programs, from situation comedies to news broadcasts, without English translation. They will be questioned by the computer to check their understanding of what was said on the video. Then, instead of emphasizing traditional vocabulary and grammar lessons, the computer will then lead them through a series of interactive lessons designed to assist learning the language in context by referring back to the situations in the video.

While this initial pilot phase is concerned with recognizing spoken Japanese, later parts of the project may develop other aspects of Japanese. Thus speaking, reading and writing will eventually be covered.

*Educational Technology Center
Information and Computer Science
University of California
Irvine, CA 92717
(714) 833-6911

INSTRUCTIONAL ENGINEERING:

A PLACE FOR A NEW SCIENCE

Richard Brandt*

University of Georgia

Engineering is defined as the "art or science of making practical application of the knowledge of pure science such as physics, chemistry, biology, etc." An engineer is defined as a "person versed in design, construction and use of engines or machines or in any of the various branches of engineering".

There are numerous models designed to aid the practitioner in the Instructional Design Process. Within each of these models is an area which could be considered the province of Instructional Engineering.

This paper presents several models of the instructional design process and describes where in these models a science of instructional engineering fits. The definition of instructional engineering is differentiated from the terms technology of instruction; instructional technology; instructional design; and instructional development. A comparison is made to the theoretical work being done by instructional scientists such as Charles Reigeluth and Frank Roberts. The parameters within which an instructional engineer can work and models demonstrating how these parameters form a structure are presented.

This engineering issue is important when considering the design of interactive instruction since such instruction must be able to stand alone without the aid of teacher intervention.

*Department of Instructional Technology
607 Aderhold Hall
University of Georgia
Athens, GA 30602
(404) 542-3810

THE TEXAS LEARNING TECHNOLOGY GROUP
COMPLETE CURRICULUM FOR PHYSICAL SCIENCE:
A PILOT STATUS REPORT

Stephen F. Brown*

Texas Learning Technology Group

The Texas Learning Technology Group's premier project, the development of a 160 hour complete curriculum based upon interactive videodisc technology, is well into its pilot phase. This presentation will detail the progress made since last year; eight chemistry discs and two physics discs will have been produced, with four more physics discs and one teacher training disc under production. One focus of the discussion will be tremendous implementation logistics involved in this total curricular approach to the use of interactive videodisc technology. Exerpts of the courseware will be shown. Of particular interest to designers is the use of non-competitive group instruction to promote cooperation and peer learning. Further discussion will focus on teacher reaction to their changing role in the classroom; that of changing from a whole-class performer to a manager of the learning process.

*Curriculum and Technical Manager
Texas Learning Technology Group
P.O. Box 2947
Austin, TX 78769-2942
(512) 467-0222

DESIGN STRATEGIES FOR INTERACTIVE VIDEODISC PROGRAMS

Geri Gay*

Cornell University

The advent of visual computing systems is reshaping the way courseware is designed, developed and used. Interactive videodisc systems have the capability to link various modes of presentation. Students are able to access video images, graphics and text, and to use these representations of material to clarify understanding.

Special skills are needed to develop interactive videodisc programs. Instruction using videodisc requires consideration of factors in learning theory, such as learner interaction, learner control, feedback, types of frames, branching, lesson structure and record keeping. These factors suggest the design of more knowledgeable teaching programs including a comprehensive representation of the student's knowledge state on which to base decision making, and for a richer form of dialogue between the student and computer programs. In this presentation, we will provide an overview of theoretical and design requirements for interactive media. We will share our experience in planning, producing, using and evaluating videodiscs in education.

These concepts will be illustrated using the IBM Infowindow system and material developed at Cornell University, including the recently developed "Murder She Computed", a participative mystery game used to teach German.

*444 Day Hall
Ithaca, NY 14853

(607) 255-7737

BASIC PRINCIPLES OF INTRAVENOUS THERAPY - AN INTERACTIVE VIDEO

Kathleen L. Guyett*

Genesee Community College

Interactive video was chosen over traditional lecture/demonstration as the format for presenting the theory and practice skills described in this project. The program was designed to teach intravenous therapy to beginning level nursing students. One of the reasons that the interactive format was chosen is because of the repetitive nature of many of the aspects of intravenous therapy, such as calculation and counting of drip rates. The interactive video allows the student the amount of "drill" he/she needs to master the skill. Another advantage of the interactive video in this case, was the ability to visually display some of the small equipment, and markings on the equipment that would not be visible to all students in a demonstration or lecture. Consistency in the presentation of material to various groups of students, and the potential for use as a remedial tool were other reasons that the interactive format was chosen.

The software used to develop the interactive video was the INTERACTIVE DESIGN AND AUTHORING SYSTEM (IDeAS) produced by OmniCom Associates.

Five specific content areas were presented in the interactive video. Each area of content had its own learning objectives, which were printed on the screen at the beginning of the section. The learner followed a typical interactive video format to learn the material and progress in the program. Repetitive skill areas such as counting drip rates included multiple situations in which the student had to demonstrate proficiency in order to continue. A college lab followed the interactive instruction where students practiced the skills they learned.

The outcomes of the use of this interactive video demonstrated it to be a viable alternative to the lecture/demonstration method of teaching basic theory related to a skill. Faculty found greater skill level in the clinical area with less students being referred for remedial practice and performance on skill evaluations improved 35-40%. Interactive video programs do take some time and effort to develop, but are well worth it if the outcomes are as positive as the ones demonstrated here.

*Genesee Community College
One College Road
Batavia, NY 14020

(716) 343-0055

C.A.S.T. - A LEVEL IV SONAR TRAINER

Tom Jones*

The University of Calgary

The Canadian Department of National Defence contracted Honeywell Canada, Ltd. to develop a part-task trainer/simulator for use in an Acoustic Operator course at the Canadian Forces Fleet Training School in Halifax, Nova Scotia. The contract called for development of a computer-assisted systems trainer (C.A.S.T.) which was to incorporate laserdisc technology. The targeted segment of the Acoustic Operator course was a unit on the process of the initial detection and classification (ID and C) of sonar echoes. This unit, which was concerned primarily with the acquisition of a complex aural discrimination skill, was previously covered in nine days of a three-week Sonar Operator course. Honeywell was responsible for the design and production of the microcomputer/videodisc (level IV) configuration which would serve as an individualzed, self-paced trainer. The rationale for the project was to test the feasibility of innovative training technologies in the Canadian military and to evaluate the applicability and potential of such technologies for future initiatives.

The Learning Systems Department of Honeywell Canada, Ltd. (North York, Ontario) carried out the project over a ten month period. The final product was a stand-alone, microcomputer/videodisc-based, individualized training system which provided 6-8 hours of interactive tutoring and testing on the classification of sonar echoes.

*Department of Educational Psychology
The University of Calgary
Calgary, Alberta T2N 1N4
Canada

(403) 220-6498

CAST: AN INTERACTIVE VIDEODISC SYSTEM FOR SCIENCE TEACHERS

James M. King,* Thomas C. Reeves, James R. Okey

The University of Georgia

With an increasing emphasis and demand upon educators to keep abreast of the latest technologies and curriculum materials, in-service training to maintain and extend teacher competencies is a necessity. Science teachers, particularly, are confronted with an overwhelming amount of information and materials designed for classroom instruction. A shortage of certified and qualified science teachers places demands on up-grading traditional teacher training materials to provide the quality science teachers needed to staff our schools in the coming years.

CAST (Computer Assistance for Science Teachers) programs, funded by a two year NSF grant, is designed to up-grade middle and high school science teachers skills in three areas, computer-based simulations, data analysis and testing. Through the use of an interactive videodisc program, study guide and supplemental computer programs, teachers are trained to incorporate computer-based simulations, data analysis and graphing programs and computer-generated testing techniques into their instruction.

Currently both an Apple and IBM version of the CAST interactive program are being field tested. The Apple version uses a relatively inexpensive videodisc system with limited video image capabilities. The IBM version is designed to operate on IBM's InfoWindow system and incorporates sophisticated screen overlay, touch screen and synthesized audio capabilities.

This presentation describes the design, production, programming and evaluation considerations involved in creating the CAST program. The program will be demonstrated using an IBM InfoWindow system and results of an on-going field test will be presented.

*Instructional Technology
The University of Georgia
607 Aderhold Hall
Athens, GA 30602
(404) 542-3810

REPURPOSING VIDEODISCS

A RESEARCH STUDY EXAMINES THE ISSUES

Carolyn S. Lay*

VANTAGE POINT Systems

The high costs of interactive videodisc technology prevent many educators from developing customized videodiscs. Nevertheless, educators continue to explore the possibility of the instructional medium in a variety of ways. It appears that many developers of interactive activities first experience their application of videodisc technology through a process which has become known as "repurposing a videodisc".

In September 1986, the U.S. Department of Education awarded VANTAGE POINT Systems a research contract to examine the repurposing process in greater detail. The specific topics about repurposing currently being studied are the:
1) Terminology associated with repurposing. 2) Instructional systems development process of repurposing. 3) Information dissemination about repurposing activities. 4) Legal implications of repurposing.

At the current stage of the study project researchers have identified:
a) Three distinct types of repurposing activities, b) three levels of information dissemination (and related development tools) which would stimulate continued exploration of this instructional medium.

Project researchers continue to examine: Upper elementary and middle school interactive videodisc applications for special education students in science; special education and science teachers' ideas to apply this technology in the classroom and legal implications of repurposing.

A summary of these research findings will be presented during this session.

*9043 Dunloggin Court
Ellicott City, MD 21043

(301) 465-6193

DEVELOPING A TREATMENT FOR CONTENT:

AN EXAMPLE USING THE INFOWINDOW SYSTEM

Mary Leight,* Mary Richter

Crawford Communications

This presentation will illustrate how the elements of content and treatment were brought together in the design of Programming Design Fundamentals, a course designed, produced and developed by Crawford Communications on the InfoWindow System.

Designing a treatment that holds the attention of the user as well as accomplishing the goals established for a program is one of the crucial points in the design of any program. Thorough task analyses and well written objectives do not necessarily translate into a treatment that is compelling or one that holds the attention of the user. The reverse can also be true; a treatment may pack a lot of excitement, but the course may not meet the original instructional purpose.

The process of choosing a treatment for a particular body of information involves three key players. The designer, producer and programmer need to coordinate efforts developing treatment. During this presentation we will discuss the creative process that was used to design Programming Design Fundamentals and how we approached problems in phases of design, production and programming.

*Crawford Communications
506 Plasters Avenue
Atlanta, GA 30324

(404) 876-8722

ARTSEARCH VIDEODISC RETRIEVAL SYSTEM FOR THE INSTRUCTION OF ART AND DESIGN

Patricia K. Mansfield*

University of Wisconsin-Madison

An authoring language is integrated with an interactive videodisc retrieval system for teaching university level courses in art and design. This approach is designed to meet the needs of instructors who use large quantities of visuals in the classroom, who teach lecture format classes and who are training students to develop a visual ability. The multimedia approach presents simultaneously information of several types including still images, video, manipulated digitized images, computer-generated graphics, sound and text.

An authoring language is integrated with an interactive videodisc retrieval system, ARTSearch, which has been implemented for the Helen L. Allen Textile Collection, owned by the University of Wisconsin. The visual database on the videodisc and the accompanying text database provide the content for the instructional modules. Software controls the player, accesses the text database and merges video, text and graphics. Presentation of sample lecture modules will demonstrate how academic disciplines can effectively utilize large quantities of multimedia visual materials in the education process, both for support of classroom lectures and for student self-study. In addition, design students are able to utilize graphic workstations equipped with videodisc players. They utilize image processing and graphics software to originate contemporary textile designs using the databases of the Helen L. Allen Textile Collection as a design resource.

*University of Wisconsin
1300 Linden Drive
Madison, WI 53706
(608) 262-2651

EFFECTS OF COMPUTER-ASSISTED VIDEO INSTRUCTION ON NURSING STUDENT ACHIEVEMENT AND LEARNER CHARACTERISTICS

Barbara J. McNeil*

Lewis and Clark State College

This quasi-experimental study was designed to compare the effectiveness of computer-assisted video instruction (CAVI) with two traditional teaching strategies (reading-only and lecture-demonstration) by observing differences in the amount of cognitive learning and psychomotor skill performance and measuring interaction effects between learner characteristics and teaching strategy among first year Associate Degree Nursing students.

Data analysis from 40 subjects revealed statistically significant group differences in ability to use a stethoscope immediately after and one month following exposure to one of the teaching strategies. Compared to students in the read-only or lecture-demonstration group, subjects who utilized computer-assisted video instruction demonstrated better skill performance one month following this teaching method. Compared to subjects in the read-only group, CAVI subjects performed better immediately following exposure to CAVI. No significant differences were found between groups on cognitive test scores, although subjects in the CAVI group scored consistently higher than either of the control groups.

Due to the small sample size, data did not support the hypothesis that more years of experience in nursing affected skill performance. Attitudes toward computers in nursing and learning style had an impact on skill performance. Subjects in the CAVI group whose learning style was classified as either concrete sequential or abstract sequential tended to earn higher scores on cognitive and skill performance measures compared to CAVI learners with different learning styles.

*Division of Nursing
Lewis and Clark State College
Lewiston, ID 83501
(208) 700-2250

DEVELOPING CBE SOFTWARE ON THE
IBM INFOWINDOW SYSTEM

George A. Reed*

University of Delaware

The Office of Instructional Technology (OIT) at the University of Delaware has been working in the area of interactive videodisc for four years and in that time has used a large variety of hardware and software systems. Indeed the diversity of incompatible systems available has limited the amount of effort that was devoted to videodisc development.

The entry of IBM into the field of interactive videodisc was seen as an opportunity to standardize development and delivery on a single, readily available system. OIT has installed 22 IBM InfoWindows Systems in its facilities for development and student use. The InfoWindow is the targeted system for several interactive videodisc applications now being developed.

The InfoWindow has many unique capabilities and not a few problems. And, as with any hardware used for CBE, the InfoWindow can only be as good as the tools available for it. So, in addition to investigating the InfoWindow itself, a part of our work has been to survey the languages and authoring systems which support InfoWindow development.

This session will discuss the capabilities and limitations of the IBM InfoWindow for CBE development and delivery and survey some of the software development systems available for it. Systems discussed will include, but not be limited to the following: IBM LS/1 Authoring System, IBM InfoWindow Pilot, TenCORE, Quest and Genesis.

*Office of Instructional Technology
University of Delaware
Newark, DE 19716
(302) 451-6672

USING INTERACTIVE VIDEO TO ASSIST READERS COMPREHEND TEXT

David Reinking,* Jiin-Tian Chyou

University of Georgia

This presentation will demonstrate an application of interactive video to reading and learning from text. The IBM InfoWindow system was used to create text that enables readers to request assistance when they experience comprehension difficulty. For example, a reader who encounters an unfamiliar word in text displayed on the screen may touch the word. The reader may then touch any of several icons to request a written definition, a video image pertaining to the word's meaning, a concordance that shows the word's use in multiple contexts, or the word's pronunciation. In addition to receiving assistance with difficult vocabulary, the reader may request the following types of assistance: 1) the structure of the passage or individual paragraph, 2) a less technical version of the original text, 3) additional background information and 4) video images that fit the content of the text.

Potential applications of this program will be discussed. These include: expanding options for learning from text-based instructional materials and enhancing readers' metacognitive skills. The presentation will also highlight how low-cost, yet sophisticated, interactive video programs can be created by using existing videodiscs, the InfoWindow system, PC/Pilot and graphics application program.

*Reading Education
University of Georgia
309 Aderhold Hall
Athens, GA 30602
(404) 354-8320

RESEARCH DESIGNS FOR INTERACTIVE VIDEO

Dent M. Rhodes*

Illinois State University

Computer aided interactive video (CAIV) as an instructional technology is in transition from "intuitional" program designs to designs which are informed by research. At the same time, the use of CAIV as a research tool in other disciplines is receiving more attention.

The nature of much CAIV research has been subject to criticism, however, and the adequacy of the "experimental" research paradigm seriously questioned (Reeves, 1986). This presentation describes an alternative paradigm for interactive video research and demonstrates how the paradigm has been and can be used to develop research designs.

The research model to be employed has been labeled the "reciprocal" paradigm (Clark and Salomon, 1986) and corresponds to the "transactional" conception of interactive video design (Rhodes and Azbell, 1985). The reciprocal research paradigm emphasizes the active role of the learner or user and the transactive interaction between learner and perceived environment.

This presentation outlines basic methodological questions to be addressed in designing CAIV research using the reciprocal paradigm and gives examples of how these questions have been answered. Among the examples will be research designs on elements of CAIV including user control, scene length, and feedback modes. Designs using CAIV as a research tool which involve interaction between user and technology will be considered, as will CAIV research in cross-cultural settings. Illustrations of these designs will also be provided.

Since the details of a reciprocal/transactional paradigm for CAIV research are still in process, questions from attendees will be encouraged. A short paper on reciprocal research design will be available.

*Department of Curriculum and Instruction
Illinois State University
232 DeGarmo Hall
Normal, IL 61761
(309) 438-5461

THE VIDEO IN INTERACTIVE VIDEO:

MAKING PICTURES SPEAK

Janise Richards*

University of Texas

The goal of this "how to" session is to introduce designers of interactive video productions to the basic principles of scriptwriting and storyboarding.

There are many different ways to organize scripts for an interactive video project. This workshop plans to address a simple, well used style that most designers and producer/directors use or have used.

During this workshop the participants will first learn the principles for each component and then apply those principles before moving on to the next component which builds upon the first.

In the first portion of the session we will demonstrate the use of outlines, good old index cards, as well as flow diagrams as some examples of methods to organize content. The participants will be asked to outline content for a simple script.

The second portion of the session will describe some of the most common errors that occur when scripts are written. A series of problems will be presented and analyzed. The participants will be asked to revise the problem scripts.

Next, the different forms used for scriptwriting, such as two column scripts, film scripts, half-page scripts, will be discussed. The participants will be asked to choose a form which most suits the content and/or work style and develop a one page script.

Visualizing the script is the next portion of the session. The participants will now have an opportunity to scribble out some visual ideas to portray the script. These visual ideas will be blended into a rough storyboard.

At the end of the workshop the participants will have examples of different styles of content organization and scriptwriting to take with them as well as a rough draft of a storyboard.

*Health Science Center
University of Texas-San Antonio
7703 Floyd Curl Drive
San Antonio, TX 78284-7896

(512) 691-6306

INTERACTIVE READING

Barbara Signer*- University of Massachusetts-Boston

Marie Gannon - Lesley College

This presentation will explain the production and features of an interactive video tape project designed to motivate youngsters to read for information. The program utilizes a game format. The object of the "game" is to get Paul Revere from Boston to Concord. This is accomplished by answering questions based on video segments and written stories that are read from booklets. All questions require the youngster to answer with sentences or phrases, thus avoiding multiple choice questions. If students do not have sufficient scores after completing the program they have the option of reading additional stories, with accompanying videos, to increase their scores. Feedback is given by a computer generated graphic of a map showing Paul's progress as well as by a narrator (a patriot). Students do not have to complete the program in one session. Their scores are stored on a disk and retrieved at the next session. The program is designed for upper elementary school through middle school students.

The program was developed in a two semester graduate course at the University of Massachusetts at Boston. 1. The concept for the script was developed by reading teachers from the Boston area and refined into a final script by a video producer. 2. The instructional design was provided by a media specialist and computer programmer. 3. The video was produced by the University of Massachusetts/Boston Media Center in conjunction with a local video producer. 4. The interfacing utilizes the BCD card and was provided through the Lesley College Media Center. 5. The computer program was written in APPLE SUPERPILOT by the professor of the course.

*I.L.T.
University of Massachusetts-Boston
Boston, MA 02125
(617) 929-8527

MAKING THE MEDIA THE MESSAGE

Lynne Stevenson*

Computer Sciences Corporation

With thousands of interactive videodisc (IVD) systems soon to be fielded, the Army has a message -- properly used, this new media is a powerful teaching tool. Thus, the Army required an efficient, effective means to teach IVD capabilities, design and production to an audience of military and civilian courseware developers, managers and video producers. The solution is the media device itself.

The lessons on the videodisc, "IVD: Exploring Its Power and Capabilities" address videodisc player, computer and video editing capabilities, as well as IVD lesson design. Frequent practical exercises simulate the process of making IVD design decisions and allow the student to see the result of his decision.

The student selects from several program levels: linear lesson or menu-driven reference manual of basic and detailed IVD information. He then interacts with either the chosen program itself or directly with the examples. These are "mini" programs demonstrating an IVD capability in one or more applications. Individualized remediation encourages the student to explore all answers in the practical exercises.

The student discovers the message concerning the power of IVD by interacting with functions and applications of the media device itself.

*Computer Sciences Corporation
6th and Jackson Street
PO Box M
Ft. Eustis, VA 23604
(804) 887-1677

SCRIPTING INTERACTIVE VIDEO LESSONS: AN APPROACH THAT MEETS THE NEEDS OF BOTH THE DESIGNER AND PROGRAMMER

Mary Stoddard Trainor,* Andrew E. Andrews, Stephen J. Long

Los Alamos National Laboratory

Following the analysis phase and the writing of the courseware design document, detailed design must be done. For interactive videodisc lessons, the conventional design method is through the use of storyboards, yet a search of the literature on interactive video instructional design revealed the following about storyboarding: The literature often provides a high-level description of the storyboard process instead of sufficiently detailed information to get started; if details are given, a very specific form is recommended that does not apply to your hardware/software or your application; and some designers advocate development of full scripts and storyboards, while underestimating the complexity of the task.

The purpose of this presentation is to provide new and seasoned interactive video designers with an approach called "scripting" which has been developed and used by the Training Research Team at Los Alamos National Laboratory. The "scripting" approach is cost effective in that it streamlines a complex process, thus eliminating redundancy and ensuring a product that is the result of an entire team's thinking.

The presentation will describe the scripting approach, provide many examples from recently completed projects, and compare this approach with other approaches to scripting/storyboarding.

*Training Research Team
Military Systems Group
Mail Stop F601, S-6
Los Alamos National Laboratory
Los Alamos, NM 87545
(505) 667-7862

VISUAL CONSIDERATIONS FOR SCREEN DESIGN

Ralph Wileman*

University of North Carolina

The visual aspect of computer-based instruction affects the learner and learning. As we design and evaluate messages in computer formats, there are five major questions we should ask ourselves: Is the visual readable? Is the visual well organized? Is the visual to the point? Is the viewer's attention directed? Is the visual compelling?

Answers to these questions are not simply "yes" or "no"; rather, they range from "yes" to "probably" to "perhaps" to "doubtful" to "no". And the range varies because of the detailed considerations regarding each question (e.g. when we ask the first question regarding readability, we must consider size, weight, figure/ground relationship, rate of display and so on for both words and images; as we answer the fourth question, we must consider the use of emphasizing devices such as typeface changes, color, shape, line, arrows, movement, texture, etc.).

The questions and their accompanying considerations are dynamically related; however, they can be considered and adjusted in isolation from one another. Attention to the many considerations that keep us answering the five basic design questions will increase the efficiency and effectiveness of our computer-based instructional materials.

*101 Peabody Hall 037A
University of North Carolina
Chapel Hill, NC 27514

(919) 962-3791

FACULTY INCENTIVES AND REWARDS FOR DEVELOPMENT OF CAI

Ronald C. Comer*

Ohio State University College of Medicine

The effectiveness of faculty incentives and rewards for developing CAI materials was studied at the Ohio State University College of Medicine. The major research questions being investigated included:

1. What incentives and rewards do college of medicine faculty and administrators perceive as being most effective in encouraging faculty to develop computer-assisted instruction materials?

2. Are there significant differences of opinion between administrators, faculty authors of CAI and faculty non-authors of CAI?

3. Is there a consensus of faculty and administrator opinion regarding incentives and rewards?

A three-round Delphi survey technique focusing on extrinsic incentives and rewards was used to collect participant opinions. An equal number of randomly selected participants were identified from each of three groups. The participants were provided with feedback on rounds two and three of the study that included descriptive statistics and participant comments on each item.

The highest rated incentives/reward factors included: 1) obtaining promotion and tenure credit for development of CAI; 2) having microcomputers located in faculty offices; 3) reducing the other teaching, research and service loads for faculty involved in the development of CAI; 4) paying royalties on the sale of CAI to faculty; and 5) providing instructional design and programming support staff.

Of equal importance, the lowest items were: 37) a departmental award for "Technical Teacher of the Year;" 38) royalties on sales paid to the college; 39) college copyright of CAI; and 40) university copyright of CAI.

The Delphi procedure did work toward achieving consensus and the results indicated that faculty and administrators are in agreement about the effectiveness of incentives and rewards to encourage faculty to develop CAI. Differences between groups occurred less than would be expected by chance and the reliability of the total sample on the third round suggested that the instrument was very consistent.

*Ohio State University College of Medicine
376 W. 10th Avenue
Columbus, OH 43210
(614) 292-6192

AN EVALUATION OF A COMPUTER CURRICULAR INTEGRATION PROGRAM

Elisabeth M. Craig*

University of Tennessee at Chattanooga

In late 1986, a Computer Curricular Integration (CCI) program was funded within the Center of Excellence for Computer Applications at this institution. The stated goals of the CCI program were to increase faculty computer literacy, to expand the usage of computers in the undergraduate curriculum and to assist faculty in developing software.

Since accountability was, and is, a critical factor, it was decided to make a campus-wide evaluation this year and repeat the process in several years. These pre- and post-evaluations are benchmarks to determine and to document both the qualitative and quantitative growth patterns. In addition, they will indicate the magnitude of the shift from mainframe to microcomputer.

Three assessment methods have been used in the evaluation process: A) A survey of individual faculty members to identify usage patterns. The emphasis is on imparting information about computer availability as well as on gathering data on current and future usage; lack of knowledge on the part of the faculty implies an area of weakness that must be addressed. B) An inventory, at the departmental level, to identify computer oriented courses, the type of equipment used (micro, mainframe), the software, number of students involved and the approximate hourly usage per student. The 1987 inventory is an extended version of one done yearly for the past four years. C) A log of student and faculty usage of the mainframe and micro clusters.

This presentation describes the development of the benchmarks; it reports on the findings of the 1987 baseline pre-evaluation. Our experiences may be helpful to centers doing a baseline evaluation or defining measures of accountability. Handouts will include a summary of the findings and examples of the instruments which were used to collect the data.

*Center of Excellence for Computer Applications
University of Tennessee at Chatanooga
413 Hunter Hall
Chattanooga, TN 37401

(615) 755-4396

COMPUTER BASED INSTRUCTIONAL DESIGN

A FEW SOLUTIONS

Carol A. Dwyer*

The Pennsylvania State University

The Computer Based Education Laboratory of The Pennsylvania State University assists faculty with instructional software development for use within the University. The design and development process utilized by the instructional development team, under the leadership of an instructional designer, includes many decisions and requires refined management techniques.

Early (and firm) decisions regarding screen design protocol and procedures which are common to all modules of all programs (e.g., prompts and help) provide the groundwork for design. Second, instructional principles such as the degree of user control over sequencing, necessary record-keeping and supplementary on-line and off-line materials must be determined for the specific project. The first steps in instructional design should include the writing of introductory screens such as the goal and the program requirements in the exact format which the user will eventually view. These early decisions further clarify the overall design plan.

Throughout the presentation examples will be given of effective, efficient and practical procedures which contribute to the successful and timely design, development and completion of a high-quality computer based instructional project.

*Computer Based Education Laboratory
The Pennsylvania State University
12 Willard Building
University Park, PA 16802

(814) 863-1555

WHY INSTRUCTIONAL SOFTWARE PROJECTS "FAIL"

Margret Hazen*

University of North Carolina at Chapel Hill

Within higher education, instructional software is frequently developed with external funding. Therefore, these projects have defined start and finish dates, have funding limits and often have objectives that need to be completed within specifications. Many instructional software projects seem to "fail" or fall short of expectations in some regards. Some projects fail more blatantly than others. Two techniques, the systems approach and the project management, aid the instructional designer to anticipate for and plan for as many potential failures or pitfalls as possible.

Both systems approach and project management sources recommend beginning with a prototype and several problems can occur during the prototype stage -- during detailed design, implementation, evaluation and revision phases. When prototype development is considered complete, several other problems can occur during full scale system implementation. Scaling up from a small to a large implementation creates problems, including the need to work with a different and differently motivated personnel. Project management sources stress the significance of the selection of the right person for project manager and appropriate management techniques.

The potential pitfalls of instructional software projects will be analyzed. Several examples will be provided and others will be solicited from the audience. It is hoped that a better awareness of potential pitfalls and problems will lead to better planning and design phases; those phases acknowledged to be essential but not always easily accomplished.

*Microcomputing Support Center
402 Hanes Hall 019A
University of North Carolina at Chapel Hill
Chapel Hill, NC 27514

(919) 962-0101

COPYRIGHT INFRINGEMENT: "LOOK AND FEEL"

IMPLICATIONS FOR SOFTWARE DEVELOPERS

Thomas Lee Hazen*

University of North Carolina at Chapel Hill

Federal copyright law protects expression but not ideas. Computer software presents a unique mix of expression and ideas, making it difficult for the courts to determine the portion that is protectable from unauthorized copying by others. While it is clear that programming code is copyrightable, there is considerable doubt as to the scope of protection for screen images. A number of recent decisions have held that similarity in the "look and feel" of the finished software can be the basis for finding copyright infringement, even in the absence of copying the source code. Basing infringement on the "look and feel", including user interface design, expands the traditional copyright law's protection of expression to the realm of ideas. The "look and feel" cases come very close to protecting functional aspects of copyrighted software. As such they present difficult problems for software creators in deciding how much they can "borrow" from the current state of the art.

Expansion of copyright protection into the realm of ideas and utilitarian aspects of software creates problems for software developers who are drawing from ideas in other programs. The presentation will examine the extent of permissible similarity and provide a checklist to help identify possible copyright infringement in connection with software development.

*University of North Carolina at Chapel Hill
School of Law
van Hecke-Weitach Hall
Chapel Hill, NC 27514

(919) 967-4303

PLANNING FOR INTEGRATED TECHNOLOGY-BASED INSTRUCTIONAL CENTERS

Joy R. Hughes*

University of Charleston

In these days of declining resources and escalating needs for expensive technology in the support of instruction, it is no longer feasible for universities to compartmentalize instructional resources. The designers of new or modified instructional facilities must plan to make maximum instructional use of present and future technology while at the same time facilitate an optimal sharing of human and technological resources among all departments requiring the new technologies - the library, the computer center, the audio-visual department and the computer-equipped classrooms and laboratories.

This presentation will share the results of an extensive investigation of universities throughout the country which have had the foresight to engage in such planning. It will discuss topics such as the identification of resources that can be shared, planning for the sharing of those resources and the role these shared resources can play in the support of instruction.

It will conclude with recommendations, drawn from the experience of several universities, on the development of a process for assisting representatives of different constituencies (the library, the computer center, the instructional technology department, the faculty and administrators responsible for computer-based instructional systems) to plan for an integrated future with maximum interrelationships among their here-to-fore separate departments.

*University of Charleston
Charleston, WV 25304
(304) 357-4702

STRATEGIES FOR FACILITATING FACULTY ACCEPTANCE OF TECHNOLOGY ASSISTED INSTRUCTION

(OR "YOU CAN LEAD AN OLD HORSE TO WATER, BUT CAN YOU GET HIM TO USE A COMPUTER?")

Harvey L. Hughett*- University of Idaho

The role of many learning programs in small schools will deteriorate unless means are developed to "increase their teaching reach" through continued development of technology and innovative instructional programs. Through the judicious incorporation of auto-tutorial courses, CAI and CAVI, the Department of Foreign Languages at the University of Idaho has been able to improve teaching efficiency and significantly increase overall enrollments.

In spite of initial reluctance by faculty to investigate the advantages of computer-based instruction, successes came about only because of a systematic (and non-threatening) plan designed to gain the confidence and support of teaching faculty.

Federal and state grants allowed for the purchase of equipment and faculty time for course production. Today, interactive video and CAI programs are used to teach undergraduate Spanish and French. The pedagogical validity of the programs has been documented.

There are thousands of media and computer professionals who proclaim the microcomputer to be a tool that can significantly improve the efficiency of teaching. Unfortunately, this same excitement is not always shared by teaching faculty and, in some cases, may be greeted with resentment and skepticism. This presentation outlines the steps for the successful integration of computer technology into an established teaching/learning program.

*Media Center
University of Idaho
Moscow, ID 83843

(208) 885-6411

FORMS, FORMS, FORMS: AIDS FOR THE INSTRUCTIONAL DESIGNER'S NIGHTMARE - MANAGEMENT!

Marilynne W. Mathias*

The Pennsylvania State University

This presentation addresses a critical issue in the design, development and evaluation of courseware - managing the process via forms. To ensure a smooth flow, the Computer Based Education Laboratory (CBEL) at Penn State created a variety of forms: Faculty Proposals, Intake Interviews, User Evaluation, Design Documentation, General Software Evaluation, etc. To eliminate reinventing the wheel, these will be shared during a discussion presentation.

In general, instructional designers receive minimal training in management skills. University faculty and other subject matter experts are unfamiliar with the design process. When inexperienced student programmers are added to the courseware development schedule, chaos could result. To keep a multi-project unit on task with a small staff, a solution was necessary. The reasonable solution included the development of forms. These forms permitted management of the courseware development process by a variety of individuals.

Prior to the development of each form, decisions were made on the critical facets of CBEL's mission. Once the significant passages were agreed upon, requests to other universities were made. Each developed Penn State form reflects the input of other university software development units plus the philosophy of the CBEL staff.

The presentation will include a discussion of the forms effectiveness to the courseware development process. Copies of each form will be provided to attendees.

*12 Willard Building
The Pennsylvania State University
University Park, PA 16802
(814) 863-2271

THE CBT DESIGN DOCUMENT

Marjorie Pomper*, Joan Saz

Applied Science Associates, Inc.

At the foundation of any successful CBT program is an effective design. The development team must understand and agree on the course design before development begins. The CBT design document provides a systematic means for the specification and communication of course design. Development team members, including the project manager, vendor/author, and subject matter expert employ the design document to ensure that content and course structure decisions are made and that the necessary agreement is obtained.

Design elements that are unique to the CBT medium are considered in the design document, along with traditional design elements that are the foundation of all training, regardless of media. Course purpose, audience description, objectives, teaching philosophy, evaluation methods and a range of estimated times for course completion should be specified in the design document. In addition, a description of the types of interaction and the frequency of interaction should be included. Information about the authoring system and which capabilities will be utilized should also be included. Finally, frame maps that specify the course structure are an important component of the design document.

Frame maps illustrate the instructional sequence, major branches and evaluation frames for each lesson. The design of control, alternate paths and interaction with record keeping (CMI) is communicated through the frame map.

*Applied Science Associates, Inc.
P.O. Box 1072
292 Three Degree Road
Butler, PA 16001
(412) 284-7378

A SMALL IVA DEVELOPMENT PROJECT

Mary Jac M. Reed*

University of Maryland

A burning question in the minds of managers of CBI and CBT projects is the question of time. How long does it take to develop software materials for the computer?

I was given an excellent opportunity to try to answer this question. In the beginning of 1987, I was given a brand new software project to manage, with new staff using a new authoring language on unfamiliar hardware. The programmers and designers were hired for this project specifically after they agreed to keep detailed logs of their time using a Time Management Utility program. Logs were to document the time devoted to the following activities: learning the new language and the new environment; meeting with coordinators, consultants and subject matter experts; designing screen displays; coding the module; revising the screen displays and the code after student use, after design and code review and in preparation for the software to be distributed to other institutions.

The project report describes the type of software development project and the findings about the development time. Listings of the times devoted to the various portions of the development process will be available as handouts. The excellent documentation that resulted from the Time Management Utility is useful in forecasting timelines for future projects and in examining ways to streamline the development process.

*University of Maryland
Office of Instructional Technology
305 Willard Hall Education Building
Newark, MD 19716

(302) 451-8161

MANAGING COMPUTER-BASED INSTRUCTIONAL SYSTEMS

A SYSTEMS APPROACH

Debra E. Schweitzer*

MetaMedia Systems, Inc.

Effective project management of computer-based instructional systems (including interactive video/audio systems) requires the coordination of a wide variety of diverse resources. Talent and technology must be applied to provide cost-effective solutions to predetermined problems. Resources, including budgets, timelines and personnel, are finite. Moreover, the inter-relatedness and dependencies of development tasks, the fusion of technology and art, and requirements for validation make the problems of CBI/IVD project managers unique.

The systems approach to management offers a paradigm for understanding complex organizations and can be effectively applied to the management issues inherent in the development of computer-based information systems. The systems approach incorporates the traditional analytic approach to management: planning, review, requirements analysis, design, implementation and evaluation. However, it also factors environmental, personnel, philosophic and technical considerations, thereby providing a framework within which a manager can integrate all aspects of a problem. After a systematic analysis of each component part, each part is synthesized into a new whole.

This presentation will discuss a systems approach management model that considers the planning, organizing, budgeting, staffing, reporting, evaluating and controlling functions of the CBI/IVD project manager. Financial, personnel, and time-related issues common to CBI/IVD development will be examined, based upon case studies developed from a cross-section of completed projects.

*MetaMedia Systems, Inc.
20251 Century Boulevard
Germantown, MD 20874

(301) 428-9160

COMPUTER-BASED AND COMPUTER-ASSISTED INSTRUCTION IN THE HUMANITIES:

A COMPARISON OF TECHNIQUES

John C. Fisher*

State University of New York

Demonstrations of interactive computer programs designed by the presenter illustrate two different uses of computer instruction in college humanities courses. One program, designed to teach English phonology, morphology and syntax to prospective teachers of English language arts in elementary and secondary schools, illustrates the techniques of computer-based instruction. The second, designed only to reinforce what has previously been taught and discussed in a British literature survey, illustrates the techniques of computer-assisted instruction.

Computer-based and computer-assisted instruction reverse the respective roles of teacher and computer. The first finds the teacher as reviewer, while the second finds the computer as reviewer.

In the academic community, and especially in the humanities, one largely finds a heated negative reaction to the first kind of instruction and only a lukewarm reception of the second. It is incumbent upon us, however, to investigate the effect of using computers to instruct and assist instruction. The computer-based program in English linguistics is in an early stage of implementation, but the program in British literature has been in use for eight years and seems to have a measurable positive effect on student performance.

*State University of New York
College at Oswego
Oswego, NY 13126
(315) 341-2153

THE DESIGN AND DEVELOPMENT OF A PC-BASED MAPICS EMULATION PROGRAM

K. Kern Kwong,* California State University-Los Angeles

Donald Cheng, Wing M. Fok, Georgia State University

The presentation discusses the project undertaken by an institute to design and develop an emulated program of the IBM's MAPICS (manufacturing accounting and production information control system). MAPICS is a comprehensive industrial information system which covers topics such as production accounting, financial analysis and sales/distribution. MAPICS runs on IBM minicomputers and is too expensive for a university to operate for pedagogical purposes. The project's products include: a set of four diskettes, an instructor's guide, a nine-units student workbook. The program covers applications such as database, production planning and scheduling, inventory management, material planning, order management and production control. The emulated program was developed by capturing the selected screens from MAPICS and by linking these screens together to emulate the real information system. The emulated program runs on IBM PC with 256K of memory. Basically, this PC-based program displays the same screen that students would see if they were using a real MAPICS. Such an emulation program has several advantages over the real system. For example, easy to use by inexperienced students, tailored help screens, inexpensive to operate by institutes, program and courseware are very portable. Currently, the program is being used in several institutions to teach production management courses in both graduate and undergraduate levels.

*Department of Management
California State University-Los Angeles
Los Angeles, CA 90032
(213) 224-2961

THE ROLE OF THE COMPUTER IN A MUSIC THEORY CLASSROOM

Liora Bresler*

Stanford University

This study investigates the impact of the use of a computer based education system on a college-level introductory music theory class. The objectives of the study were the following: 1. to assess how and to what extent the computer-based curriculum was integrated into the operational curriculum; 2. to locate students' and teacher's barriers to using the computer; 3. to identify students' areas of success and difficulty as well as their individual choices at the computer and their strategies in solving musical problems, and to capture students' experiences as they influence their learning of music, both at the computer and in the classroom environments.

This study, by focusing on processes and examining events in their full complexity captured important aspects of the use of computers for education, which have been overlooked in previous studies. In particular, the findings from this study highlight the importance of the context in which the computer is embedded: the operational curriculum and the computer's integration into it.

*57-D Escondido Village
Stanford, CA 94305

(415) 494-3554

COMPUTER SIMULATED RHYTHMICAL TRAINING

Jon-Chao Hong*, Der-Shen Cheng

National Taiwan Normal University

In Taiwan, while most students learn musical rhythm, their teachers often play and demonstrate the rhythm of that instrument. Yet, every teacher has his own motor response problem, which results in a non-standardized or irregular rhythm. Therefore, students cannot be guaranteed to obtain a good rhythmical motor skill.

In order to solve the above problem, the software is developed to simulate the practice of musical rhythm, based on the function of the frequency clock in the computer which will reflect and control the rhythmical reaction on the screen, and judge the responses from users. The users can use the joy-stick to practice the rhythm according to the signals shown on the screen.

Basically, this software is developed for K-7 or K-8 students on the Apple II computer to practice the rhythm of orchestral instruments. This software consists of 12 units (lessons), and each unit has two questions for users to practice. Since there are five tries in each unit game, students have good chances to complete that practice unit.

Some other traits of this software include: 1) the rhythmical speed can be adjusted according to the user's skills, 2) the precision level of manipulation (degree of acceptance of error allowance) can also be chosen according to student's familiarity in operating the joy-stick.

*Department of Industrial Education
National Taiwan Normal University
Ho-Ping E. Road
Taipei R.O.C.

AN INTELLIGENT TUTORING SYSTEM IN HARMONY

Linda B. Sorisio*

IBM Corporation

The goal of our research project is to design and develop an Intelligent Tutoring System in Harmony that will provide flexible, individualized instruction for the university-level music major. The system will be student controlled and utilize several musical teaching strategies. These teaching strategies will be based on psychological learning theories as well as the practices used successfully by human teachers of music theory.

An initial system was developed, utilizing an expert systems approach, that teaches musical intervals. The inference engine of the expert system infers the strengths and weaknesses of the learner, tailors problems that will be appropriate for the level of the learner, and generates diagnostic advice. Communication between the expert system and procedural programs integrates graphics and sound into the tutoring system. Project goals, philosophical and pedagogical foundations, system designs and future directions of research are discussed.

*IBM Corporation
Los Angeles Scientific Center
11601 Wilshire Boulevard
Los Angeles, CA 90025

(213) 312-5563

USING SPECIAL GRAPHIC EFFECTS IN INSTRUCTION

Susanne M. Behe, Sandia National Laboratories

Bradley K. Schaublin, Norman C. Higgins*

Arizona State University

Authors of computer-based instruction have many special graphic effects that can be incorporated into their programs: text highlighting, multiple window screens, library graphics, animation and special type fonts. When used appropriately, these graphic effects can facilitate learning and can enhance the program's appeal for learners and instructors. When used inappropriately, the graphics can be distracting and a hindrance to learning.

In this presentation several special graphic effects will be described and reasons for using them in instruction will be discussed. Programming procedures used to create the effects in the PC PILOT authoring language will be described.

Examples of special graphics effects in an interactive video program developed for Sandia National Laboratories will be used to illustrate the presentation. Segments of the instructional program designed without the graphic effects will be contrasted with segments in which graphic effects have been used.

*1826 East Alameda
Tempe, AZ 85282
(602) 965-3651

INTEGRATION OF CD-ROM AND OPTICAL VIDEODISC VIA PC PILOT IN A MEDICAL APPLICATION PROGRAM

Richard J. O'Connor*

Lehigh University

This presentation will describe the design and development of an individualized medical education program in tropical medicine that incorporates interchangeable access to specific topic tutorials as well as current bibliographic citations from the National Library of Medicine data base, MEDLINE. The authoring language, PC PILOT, is used to direct the program, which consists of an IBM PC/XT interfaced with a Pioneer LD-V6000 videodisc player and a Philips CM-100 compact disc player. The basic intent of the program is to demonstrate that microcomputer-aided instruction is now able to integrate conventional database search within a self-study program or, alternatively, to allow the learner to undertake a tutorial on a specific topic while undertaking a bibliographic search.

The presentation will be illustrated with 35mm slides and a short videotaped demonstration of the teaching/learning system. Distribution of a short paper describing the details of the program in more depth, as well as a listing of current references relating to integration of CD-ROM/VIDEODISC technology in educational settings will be included.

*Education Technology Program
Lehigh University
Bethlehem, PA 18015
(215) 758-4794

SOUPED-UP SUPERPILOT PROGRAMMING

Marianne E. Yoder*

Beginning SuperPILOT authors are soon confronted with the space limitations imposed by the 22 block lesson module format of SuperPILOT, and the availability of only 144 blocks on the disk. Authors needing student record keeping must deal with SuperPILOT's disk-space-eating keep statement. Also annoying is the relative slowness of the program execution of link, compute and jump statements.

However, there are programming techniques which can ameliorate these limitations. It is possible to condense the amount of code needed for each lesson module through the use of variables and subroutines. The apparent speed of the lesson presentation can be increased by engaging the learner in an activity during the compute and link execution. The actual speed of execution of a jump statement can be increased through the proper placement of use routines and labels. Program length can be increased by utilizing both sides of the disk. Student record files can be created through file handling techniques which do not gobble up disk space. These techniques will be demonstrated in this presentation.

*10590 E. Placita Diego
Tucson, AZ 85748
(602) 885-4858

U.S. ARMY NEED FOR PLATO-DELIVERED COURSES

Clinton L. Anderson*

Servicemembers Opportunity Colleges

The United States military services (particularly the U.S. Army) have established a network of PLATO operational mainframes and have numerous PLATO terminals in work places and learning centers. These terminals are being used, for the most part, during on-duty time for military training and education directly related to military job performance. During off-duty times, particularly during the evening hours, many of these terminals are free for soldiers and, in some instances, for military family members on a space available basis, to participate in voluntary education offerings such as college courses.

The U.S. Army Training and Doctrine Command has initiated an action to have regionally accredited educational institutions offer courses for academic credit using the Government-owned Joint Computer Based Instruction System. Educational institutions that have developed and/or use PLATO delivery method college courses are being requested to indicate whether interest exists within those institutions to offer such courses on this system. Educational needs of U.S. Army service members and their family members are discussed by the Servicemembers Opportunity Colleges representative. Questions from PLATO users regarding hardware, software and courseware are addressed by the Army's Training and Doctrine Command spokesperson.

*Servicemembers Opportunity Colleges
Suite 700, One Dupont Circle N.W.
Washington, DC 20036

(800) 368-5622

A REPORT ON THE DEVELOPMENT OF A STATE-WIDE CBE NETWORK IN MAINE USING NovaNET(sm)

Owen F. Gaede*

University of Maine

At the 28th Annual ADCIS International Conference in Washington, D.C., Don Bitzer of the University of Illinois announced plans to develop a national PLATO-like network called "NovaNET". NovaNET is a new system which uses satellite transmission to cut communication costs and which promises to enable thousands of users to simultaneously access the same CBE system.

As a result of that announcement, the University of Maine decided to become one of the first large-scale users of NovaNET in a plan to implement a state-wide CBE system. The goal in Maine is to make NovaNET access a local call from anywhere in the state, and to use NovaNET as the basis of a wide variety of CBE and informational services, not only for the universities of the state, but for public schools, business and industry and the general public.

This topic presentation will be an up-to-date status report covering the first nine months of that project. The presentation will include an outline of the milestones to date in developing the network, an analysis of actual costs incurred and a report of system reliability and performance. In addition, a timetable of planned events during the next twelve months will be discussed. The presentation will be designed to be of particular value to persons who wish to consider NovaNET as an alternative to other forms of network-based CBE systems.

*University Innovations
Alumni Hall
University of Maine
Orono, ME 04473
(207) 581-1582

TEACHING EXOTIC LANGUAGES WITH PLATO

Celia D. Kraatz,* James M. Kraatz

University of Illinois

Advanced university students in some fields may need to learn languages which are rarely taught and for which teachers or native speakers are unavailable. This need is usually met by self-instruction courses using text books and audio cassettes. Of the four language skills of listening, speaking, reading and writing, text book and tape methods have some success in teaching the passive skills of listening and reading. But the active skills of speaking and writing are usually not very well learned because there is no teacher or other monitor to evaluate the student's practice. The PLATO system and a computer-controlled random access audio device were used to deliver a course in Afrikaans. The passive skills of listening and reading were taught as in conventional self-instruction courses with aural material presented via the random access audio device and reading material presented in the text book and via PLATO. For the active skills, student writing was monitored by the computer through writing exercises. Speaking skills were practiced by the student listening to the recorded model voice, and then recording his own speech and comparing it to the model as many times as required to reach the level of proficiency desired. The use of the PLATO system and a computer-controlled random access audio device gives the student the same practice in the passive skills as does the textbook and tape method, but in addition permits monitoring of the active skills by the computer and the student.

*CERL - 252 ERL
University of Illinois
103 South Mathews Avenue
Urbana, IL 61801

(217) 333-2375

CERL'S "COLOSSAL" COMPUTER-BASED EDUCATION SYSTEM

Sam A. Falk Milosevich*

University of Illinois at Urbana-Champaign

The Computer-Based Education Research Laboratory (CERL) at the University of Illinois at Urbana-Champaign (UIUC) has more than 25 years of experience in developing and applying computer technology for educational purposes. CERL's new "colossal" computer-based education system, called NovaNET(sm), is designed to support thousands of active users at low cost by use of modern computer and satellite communications technologies.

A large catalog is available, representing over 9,000 hours of material of which about 6,000 have accredited classroom use. Access may be via personal computers or CERL's low cost terminal. Inter-network connections will provide electronic communication with non-NovaNET systems.

NovaNET's large-scale communications system will support the equivalent of 4,000 simultaneously-active CBE users on one "T1" satellite channel with an aggregate data rate of 1.544 Mbps and a number of telephone lines with primary reverse channel aggregate data rates of 9.6 Kbps each.

One satellite down-link may serve any number of users. One return phone line may be shared by 300 users. Distributed across 300 terminals in a small geographic area, communications costs per month per terminal port should be approximately $10.00 plus local phone charges.

NovaNET(sm) is a service mark of University Communications, Incorporated.

*Computer-Based Education Research Laboratory
252 Engineering Research Laboratory Building
University of Illinois at Urbana-Champaign
103 South Mathews Street
Urbana-Champaign, IL 61801
(217) 333-6210

THE EFFECT OF IMMEDIATE AND DELAYED FEEDBACK ON RETENTION IN A COMPUTER-BASED TRAINING PROGRAM

Susan D. Moisey*

Grant MacEwan Community College

In this study, CAI modules were coupled with CMI testing and feedback strategies to explore the Delay-Retention Effect.

As part of a dissertation research project, 171 members of the Edmonton Police Department received training using eight modules of the Criminal Law Program. Modules were created using the OMNISIM authoring system, extensively field tested and delivered by on-site PLATO terminals. A variety of testing and feedback conditions were implemented using PLM software. One-half of the subjects received immediate by-item feedback, the other half received 24-hour delayed feedback. This paper compares the pre-test and post-test scores of members within each feedback group.

*Grant MacEwan Community College
11030 107th Street
Edmonton, Alberta T5J 3E4
Canada

(403) 441-4879

DEVELOPMENT OF PLATO USAGE AMONG FACULTY

Deborah Ouellette*

California State University

The number of faculty and student users on the PLATO system at California State University-Sacramento, has increased significantly since it was first installed in 1983. Since that time, on-campus usage has grown beyond 1,200 students and 148 instructors, actively using PLATO for remediation, augmentation of courses, college credit and courseware development.

At the outset, accommodations were arranged for five PLATO terminals connected to the University's Cyber 170-730 mainframe computer in the University computer lab. The nature of the project was exploratory with an overall objective to improve teaching and learning at the university.

With the increasing positive response of both the faculty and students, the PLATO project was accepted and funded. At present, the PLATO learning lab consists of 10 terminals with other smaller satellite labs located in various departments on the campus. In addition software packages have been purchased from other PLATO sites for enhancement of the system along with the implementation of peripheral devices such as sound synthesizers and interactive videodisc.

Recruiting faculty who would provide an organized student usage of PLATO was the key to the successful growth. The means to achieving such growth was through department specific advertisements and workshops/seminars that displayed to the instructors that PLATO would be a time saver and another useful teaching tool that could be utilized with the classroom experience.

Techniques for recruiting, training and stimulating continuous interest for faculty without monetary incentives will be discussed along with special training tools used to minimize the time of the training consultant.

*6516 Deaderick Court
Orangeville, CA 95662
(916) 278-7632

THE USE OF KNOWLEDGE-BASED MICROWORLDS
IN AN INTELLIGENT TUTORING SYSTEM FOR PHYSICS

Stephen F. Brown*

Texas Learning Technology Group

The intelligent physics tutor ARPHY II, developed by the author, has received a modification, allowing the system to place the student into a microworld whereby concepts and principles may be isolated and altered, either by the system or by the student. By allowing the student to view various phenomena of physics through the multitude of angles available with the microworld approach, it is hypothesized that the student will receive a more solid foundation of physics knowledge.

The knowledge base of ARPHY II is being kept separate from the teaching strategy. The teaching strategy module keeps track, at a high level, of student use and misuse of rules within the knowledge base (also called the expert module). Among the microworlds available to the student are the manipulation of gravity and air resistance of projectile motion, and the manipulation of various object masses during force interactions. Sample student interactions with ARPHY II will be shown.

*Texas Learning Technology Group
P.O. Box 2947
Austin, TX 78767

(512)467-0222

SIMULATION IN PROGRAM MANAGEMENT EDUCATION

Forrest C. Gale*

Defense Systems Management College

Computer-driven scripted scenarios are software sequenced and simulated processes in which students act and interact individually and in small groups with each other, with faculty and with technology resources to achieve learning goals and objectives. A recent DSMC research program has focused on the application of such advanced, state-of-the-art tools to simulate a real world defense acquisition program office environment. This paper reports on recent work in planning these high technology interventions, in designing and developing the resultant products, and in prototyping and testing the target simulations in a newly developed "automated" laboratory classroom facility which is unique in the field of education. Design artifacts of the simulated defense acquisition management environment are described in considerable detail, as is the psychosometric architecture of the prototype scenarios and its meaning and potential for impact on the methods and processes currently invoked in real world service program offices.

*Defense Systems Management College
Laboratory Department
Fort Belvoir, VA 22060-5624
(703) 664-6811

FEEDBACK AND TRANSFER WITH AN INSTRUCTIONAL EXPERT SYSTEM

Scott Grabinger,* University of Nebraska

Joellyn Pollock, Instructional Innovations

One of the major issues in instructional design is "transfer of training", both near-term and far-term. The assumption is frequently made that success in near transfer implies equal success in future, or far, transfer; however, success in one is usually achieved at the expense of the other. This problem of transfer is influenced by type of task, student aptitude and instructional method. In tasks containing both declarative and procedural objectives, a research study was designed to measure the effectiveness of feedback provided by an expert system on the transfer of media production and production evaluation skills. The expert system is, in essence, a self-administed fault-analysis system. On the basis of responses to questions asked by the system, a judgment about the quality of work with suggestions on how to remedy faults is provided. Through use of the expert system, students learn the evaluation skills used to analyze their own work. It is hypothesized that the expert system will encourage students to take a more "mindful" approach to their work by teaching, in a systematic way, the application of evaluation techniques to media productions and, thus, facilitate both near and far transfer of both the production and evaluation skills.

*1930 Pacific Drive
Lincoln, NE 68506

(402) 438-8380

THE EFFECTS OF ABILITY BASED GROUPING STRATEGIES ON LEARNING FROM THE COMPUTER

Simon Hooper,* Michael J. Hannafin

The Pennsylvania State University

Both criticisms of the computer for providing a sterile working environment, and the lack of availability of computers in the schools have encouraged thinking into alternative classroom management strategies for computer learning. One such strategy involves student grouping. The purpose of this presentation will be to summarize the existing research in the effects of grouping models on computer learning and to describe and illustrate a case study where subjects will be assigned to different levels of group learning based upon the degree of homogeneity and ability of the group. It is expected the study will indicate that homegeneous grouping results in significant improvement in learning for high ability students, and the progress of low ability students is significantly impaired. Students in high ability groups are likely to draw strength from others with well established learning styles and low ability students are likely to flounder in an environment that lacks successful learning strategies.

*625 Easterly Parkway
State College, PA 16801
(814) 863-4291

APPLYING AI TO INSTRUCTION

Greg Kearsley*

Park Row Software

Artificial intelligence refers to a set of programming techniques that allow "smart" programs to design, i.e., programs that understand what they are doing and can make inferences and associations. In the context of instruction, the application of AI means designing instructional programs that teach better, or, help a teacher teach better.

At the present time, there are two different ways of applying AI to instruction: intelligent tutoring systems and expert systems. Intelligent tutoring systems are programs that understand the content/task they teach. They typically take the form of coaches or diagnostic programs. The programs to be demonstrated in this session are MicroSEARCH which demonstrates heuristic search in mathematics and MicroPROUST which is a diagnostic tutor for Pascal.

Although intelligent tutors sometimes resemble traditional instructional programs, they are designed quite differently, Key elements are a knowledge representation, teaching and error rules and a student model. Intelligent tutors are usually implemented using AI languages such as LISP or Prolog (although this is not essential). There are many versions of these languages now available for personal computers.

Expert systems are programs that incorporate the rules used by experts to make decisions in a specific domain. Expert system programs allow users to solve problems in that domain by asking questions and providing advice based on the answers given. When used for educational applications, expert systems can provide hands-on practice in a course or seminar. Students use the expert system to solve case studies and in doing so learn the decision rules used by experts.

There are many inexpensive expert system generators available for personal computers. These generators resemble authoring systems and allow someone with no programming experience to create an expert system in a short amount of time. This makes it possible for instructors to develop expert systems to use in their classes with no need for outside help.

*Park Row Software
1135-C Garnet Avenue
San Diego, CA 92109
(619) 581-6778

CONDITIONS AFFECTING THE EFFECTIVENESS OF ANIMATED AND NON-ANIMATED DISPLAYS IN COMPUTER BASED INSTRUCTION

Dan Klein*

The University of Georgia

The presentation addresses the issue of graphics in instruction. It is suggested that the inconclusive results, typical to that area, are connected to inadequate conceptual/theoretical framework.

Such a framework is proposed. It regards the content and the student's cognitive system to be the determinants of the presentation (i.e., the graphic display). Thus the more specific the definitions of content and cognitive process are, the more accurate the presentation's prescription will be. Relating to the theoretical framework, it is suggested to analyze the instructional process in terms of information processing theory, and to use cognitive load as an indicator of display effectiveness.

Using the suggested conceptual/theoretical framework, an analysis of graphic displays was done which specified the conditions under which animation is effective. A study was conducted in which two types of graphic displays were presented to subjects in either animated or still mode. As predicted, animation reduced the time required to solve one type of problem (previously specified by the analysis), but increased the time required to solve the second type.

*607 Aderhold Hall
College of Education
The University of Georgia
Athens, GA 30602
(404) 542-4038

TOWARD THE REDEFINITION OF KNOWLEDGE AS DESIGN: IMPLICATIONS FOR THE DEVELOPMENT OF COMPUTER-BASED LEARNING ENVIRONMENTS

Vincent E. Lasnik,* Frederick C. Whiteman

The Ohio State University

Recent innovations in microcomputer technologies have eclipsed earlier models of knowledge as information and learning as a passive, linear, discreet and short-term process. Emerging theories of knowledge representation, multiple intelligence, parallel processing, metacognition and generative learning have recognized the active, integrated, internally-mediated processes of skill and concept acquisition.

The cognitive movement in instruction contends that learners construct meaning as they approach new content domains and problem-solving tasks and thereby design the representations of their own knowledge, often unconsciously. Since there are diverse individual approaches in the ways information is perceived, organized, retrieved and generally utilized, instructional systems need to reflect a complex, dynamic model of the learning process.

Computer-based multiple learning environments designed from a cognitive model facilitate learners to explore their own thinking processes, and to deal with ambiguity, recognize underlying assumptions and search for commonalities among data. Such systems assist the learner to use strategic tools including analogical reasoning, lateral thinking and set breaking and emphasize the essential nature of scientific discovery: the motivation to generate and confirm hypotheses in the search for causal understanding.

The presentation describes how contemporary cognitive science research can be integrated within powerful, computer-based learning environments to act as a catalyst for generating deep-knowledge structures. Critical features of relevant instructional design systems will be outlined with operational examples and practical courseware applications will be reviewed.

*2333 Neil Avenue, No. 4
Columbus, OH 43202

(614) 294-8167

THE COMPUTERIZED ADVISORY SERVICE FOR HORTICULTURE (CASH):

EDUCATING THE AGRICULTURAL COMMUNITY IN THE FIELD

Vincent E. Lasnik*

The Ohio State University

Within the last decade, the development of microcomputer-based decision support tools for the troubled, remote, independent agricultural community has become increasingly significant. Created through a grant by the Kellogg Foundation, the Computerized Advisory Service for Horticulture (CASH) Project at The Ohio State University has designed a complimentary series of simulations and tutorial programs to improve the acquisition of both farm management skills and conceptual knowledge in basic market analytics.

CASH has attempted to integrate the diverse fields of horticulture, entomology, agricultural economics and computer-based instruction in the implementation of a high-quality software decision support system to promote the systematic analysis and long-term management of successful rural cash crop enterprises. The CASH MARKET MODEL simulation is an exemplar of the next generation in intelligent farm-related software. MARKET MODEL is an easily used, deceptively powerful, multileveled, windowed laboratory environment in which growers can experiment with variable costs, qualities, quantities, prices and risks in the projected production of their commodities. Other programs, such as MICROCROP, develop and expand systematic knowledge of microeconomic principles and risk management strategies utilized in the MARKET MODEL simulation.

The presentation will review the CASH Project, including descriptions and examples of the CASH instructional software programs. Suggestions for future research and development in the field of farm-related courseware will be outlined and discussed.

*2333 Neil Avenue, No. 4
Columbus, OH 43202
(614) 294-8167

COMPUTER-BASED PRECOLLEGE ECONOMIC EDUCATION: AN IMPERATIVE NEED FOR COURSEWARE RESEARCH AND DEVELOPMENT

Vincent E. Lasnik*

The Ohio State University

Indicative of the broad yet substantial improvement across the American public school system, economic education has become an increasingly fundamental part of both the general core and college preparatory curricula. Correspondingly, there is an emerging critical need for the development of high-quality computer courseware in the area of precollege economics, and despite the dedicated efforts of the Joint Council on Economic Education, the present state of economics courseware on the secondary level remains unsatisfactory.

Much of the present courseware in precollege economics education can be characterized as poorly conceived, unimaginative, and extremely rule-oriented, failing to utilize the unique potential of sophisticated technology-based instructional systems. The next generation of these systems will apply models from the knowlege-base of cognitive learning theory and support the intelligent design of interactive simulations and tutorials, the creative use of powerful databases and spreadsheets, the animation of conceptually-rich high-resolution graphics, charts, maps and diagrams, and the carefully produced realism of engaging interactive video projects.

The presentation will review the current state of courseware in economics education, including descriptions and examples of both existing courseware and new programs currently under development. The growing body of research investigating the effects of diverse technology-based instructional strategies on concept acquisition in economics will also be examined. In addition, the presentation will outline guidelines for both the practical application of this research, and directions for further research and development in technology-based economic education. Finally, alternatives for future computer-based and interactive videodisk applications within the precollege economics curriculum will be proposed and discussed.

*2333 Neil Avenue, No. 4
Columbus, OH 43202

(614) 294-8167

EVALUATING COURSEWARE FOR INSTRUCTIONAL EFFECTIVENESS

Cindy B. Leshin*

Arizona State University

Formative evaluation is a powerful tool that can be implemented in the development of computer-based training models and computer-based instructional programs. Substantial amounts of money are wasted on the purchase of educational and training software which soon prove to be ineffective. Empirically validated techniques for the design and development of educational software would help to prevent economic waste and would perhaps lead to a more favorable acceptance of computer-based instruction/training.

The application of experimental research techniques to the validation of models and courseware has been a rare activity. Traditional designs require a considerably larger number of instructional developers (i.e. experimental subjects) than even the most affluent model maker can afford. Formative evaluation helps to resolve the problem of the need for a large population.

Formative evaluation is an effort to make a data-based analysis of the relationship between a training or instruction course and a learning effect and to use that analysis in the revising of the training or instruction when the relationship is not satisfactory.

Eight evaluation categories were developed based on the concepts of interactive, responsive and effective courseware. Standards for each category were determined for comparison. These standards describe different levels of performance for each category; examples are included for comparison.

The resulting data can be used to judge the efficacy of models on the basis of its power to shape and control the behavior of instructional/training designers. Courseware can be evaluated for effectiveness and the analysis data used for revisions.

*6101 North 1st Place
Paradise Valley, AZ 85253

(602) 840-5028

LANGUAGE AND THINKING:

A COMPUTER-MEDIATED CRITICAL THINKING COURSE

Denise E. Murray*

San Jose State University

Recent theory and research in the educational process have focused on collaborative strategies as a means of instruction; on the importance of providing real world tasks in instruction; and on the uses of literacy. Computer-mediated communication (CmC) provides a tool that facilitates all three of these foci.

This report describes a San Jose State University course in critical thinking that uses one form of CmC, electronic mail (E-mail). Student groups collect real world data in a number of different discourse domains: television advertising, televised news broadcasts, print advertising, print news reporting, class lectures, and textbooks. Groups then enter their data into files and analyze the data collaboratively through E-mail. The analyses focus on presenting evidence, establishing proof and constructing arguments. E-mail allows groups of students to enter and analyse their data asynchronously. Because their computer conversations are stored in files, students can print them out and carefully evaluate their own reasoning processes in the light of logical principles. Students' own extended argumentation becomes another data set for analysis.

This report then discusses the advantages of using E-mail as an instructional adjunct, based on results from a current research project investigating the use of E-mail in this course.

*7054 Calcaterra Drive
San Jose, CA 95120
(408) 277-2828

THE COMPUTER AS "PERSONA" IN CBI

Henrietta N. Shirk*

Northeastern University

This presentation explores the different roles (personas) which are projected by the computer in computer-based instruction (CBI). It examines these roles from the premise that, in developing computer responses in the CBI environment, the developer creates a "persona" which is that of a teacher responding to a student.

The purpose of the presentation is to heighten awareness of CBI developers to the ways in which their personal assumptions about the teacher/student relationship are reflected in the CBI which they create. These assumptions are most often based on the developer's own past educational experiences, as well as on individual's teaching style preference.

A taxonomy of computer personas is described (with accompanying examples from existing CBI) and the relative strengths and limitations of each persona are discussed. The classification of personas includes: The Authority (Behaviorist Orientation); the Expert (Structuralist Orientation); the Facilitator (Functionalist Orientation); the Counselor (Humanist Orientation).

A theory for creating an effective computer persona is suggested, which involves an adaptation of the four categories from the taxonomy to learning task analysis. The key characteristics of a successful computer persona include adaptability, believability and consistency. Summary recommendations are made for further research in matching computer personas to human learning styles.

*145 Forest Street
Haverhill, MA 01830

(617) 373-1320

WHEN CBT IS THE CHOICE BUT NOT THE SOLUTION

John Vukelich*

Ames Research Center

Many Instructional Technologists are probably faced with the same dilemma that we were; the need for a timely solution to an instructional media problem, with a number of unique variables that make the viable solution differ from the "preferred" solution. For example, in our initial media selection process we determined Computer-Based Training (CBT) to be the most effective solution to our instructional media problem in terms of design, delivery and overall cost-effectiveness. However, further in-depth evaluation proved this medium to be a non-feasible solution.

The main reason for this non-feasibility of CBT in our organization, and many organizations in the field today, is due to the lack of "off-the-shelf" systems that are compatible with current, state-of-the-art computer hardware and software technologies. This is especially true in an environment such as ours, a supercomputing environment, where state-of-the-art technology is the standard, rather than the exception to the rule.

In this session we will outline and describe the process of developing a "good" alternative solution to such a problem. We will also discuss some of the problems associated with this alternative, and what can be done to limit or prevent these problems.

*Ames Research Center
MS 258-6
Moffett Field, CA 94035
(415) 694-4338

PROCEEDINGS INDEX

A Page
Aderet, Avshalom 287
Aljadir, Letta P. 117
Allen, Brockenbrough S. 7, 295
Anderson, Clinton L. 376
Andrews, Andrew E. 354
Arnold, Jean 325
Athey, Susan 196
Attala, Emile 201
Atwater, Martha R. 291

B
Baur, Greg 174
Bayley, Garland 234
Beardslee, Edward C. 336
Becea, Lee 320
Behe, Susanne M. 372
Berkowitz, Melissa 292
Berlin, Donna 276
Blumenfeld, Gerald 317
Bork, Alfred 337
Boyer, Linda A. 311
Brandt, Richard 338
Brebner, Ann 1
Bregar, William S. 234
Bresler, Liora 369
Brittain, William H. 314
Brown, Stephen 339, 382
Brown, Gary 41

C
Cartwright, Phillip G. 322
Centini, Barry A. 268
Cheng, Der-Shen 370
Cheng, Donald 368
Cheyer, John 335
Christinaz, Daniel 147
Chyou, Jiin-Tian 349
Clark, Bill 312
Clark, Beth 312
Comer, Ronald C. 356
Craig, Elisabeth M. 357
Culley, Gerald R. 229

D
Dalton, David W. 55, 60, 62, 293
Deatsman, Gary A. 65
DeJonge, Jacquelyn 334
DeMarsh, Joseph 96
Dwyer, Carol A. 358
D'Souza, Patricia V. 132

E
Eastmond, Daniel V. 294
Emme, David 292
Epstein, Marsha 201

F Page
Farley, Arthur M. 234
Fedorchak, Gaye V. 239
Ferguson, Leslie R. 7, 295
Finkelstein, Michael W. 185
Fisher, John C. 367
Fok, Wing M. 368
Friedrich, Dan G. 282

G
Gaede, Owen F. 377
Gale, Forrest C. 383
Gannon, Marie 352
Gay, Geri 340
Geasler, Michelle 14
Goodrum, David 293
Grabinger, R. Scott 384
Gresham, Shirley F. 7, 295
Grimm, Floyd M. III 373
Guyett, Kathleen 341

H
Handler, Barbara Hershey 18
Hannafin, Michael J. 60, 62, 385
Harshbarger, Susan 121
Hathaway, Irene 329
Hayes, Joan Lauer 23
Hazen, Margret 359
Hazen, Thomas Lee 360
Heatley, Randall 329
Higgins, Norman C. 312, 316, 372
Hiranyavasit, Chairat 207
Hirschbuhl, John J. 30
Ho, Co-Nghiep 163
Holyoak, Arlene 330
Hong, Jon-Chao 370
Hooper, Simon R. 60, 385
Hughes, Joy R. 361
Hughett, Harvey L. 362

I
Iglesias, Gail 325

J
Jelden, D. L. 70
Jensen, Dorothy 331
Johnson, Lynn 185
Johnson, Patricia-Damon 107
Jones, Tom 342
Jorczak, Robert L. 135

K
Kapp, Ann B. 332
Kato, Hiroshi 142
Kearsley, Greg 296, 386

PROCEEDINGS INDEX

Page

King, James M. . . . 343
Klein, Dan . . . 387
Knirk, Frederick G. . . . 147
Kottkey, Diana . . . 101
Kraatz, Celia D. . . . 378
Kraatz, James . . . 297, 378
Kundu, Mahima R. . . . 298
Kwong, Kern . . . 207, 368

L

Lasnik, Vincent E. . . . 388, 389, 390
Lawson, John D. . . . 88
Lay, Carolyn S. . . . 344
LeDoux, John C. . . . 304
Leight, Mary . . . 345
Leshin, Cindy . . . 303, 391
Levin, Jacques . . . 268
Levin, James A. . . . 319
Levy, Lynne . . . 333
Liflander, Veli-Pekka . . . 245
Lohn-Wiley, David . . . 299
Long, Stephen J. . . . 354
Losleben, David R. . . . 37

M

Mackall, Phil . . . 323
Macmillan, Stuart . . . 292
Malcolm, Ted . . . 317
Maner, Walter . . . 256
Manna, Francine . . . 300
Mansfield, Patricia K . . . 346
Marlino, Mary R. . . . 326
Mathias, Marilynne W. . . . 363
Mattoon, Joseph S. . . . 253
McChesney, Ann S. . . . 333
McFarland, Mary . . . 101
McFarland, Thomas . . . 101
McNeil, Barbara J. . . . 347
McPherson, Mary Lou . . . 329
McRae, Pamela J. . . . 75
Medal, Mary . . . 327
Merrill, John A. . . . 301
Merrill, Zandria . . . 299
Miller, Phyllis Bell . . . 334
Milosevich, Sam A. Falk . . . 379
Mitchell, Frederick R. . . . 373
Moisey, Susan D. . . . 380
Moon, Seok H. . . . 207
Morimoto, Yasuhiro . . . 142
Morrison, James L. . . . 124
Morse, Shelley . . . 304
Muller, Elaine . . . 333
Mulligan, Patricia A. . . . 313
Murray, Denise E. . . . 392

N Page

Nagy, Monica . . . 328
Nantz, Steven E. . . . 239

O

Okey, James R. . . . 314, 343
Ouellette, Deborah Ann . . . 381
O'Connor, Richard J. . . . 374

P

Patterson, Richard L. . . . 302
Percy, Jolanta . . . 297
Perkins, Miriam L. . . . 14
Perrault, Marieanne . . . 320
Peters, Chris L. . . . 314
Pigford, D. V. . . . 191
Pineda, Mike . . . 320
Pollock, Joellyn . . . 303, 384
Pomper, Marjorie . . . 310, 364

Q

Quinby, Barbara F. . . . 335

R

Ragan, Stephen . . . 101
Reed, George A. . . . 348
Reed, Mary Jac . . . 365
Reeves, Thomas C. . . . 326, 343
Reinking, David . . . 349
Rhodes, Dent . . . 350
Richards, Janise . . . 351
Richter, Mary . . . 345
Rieber, Lloyd P. . . . 263
Riley, Mary . . . 304
Roberts, Frank C. . . . 135
Robinson, Steven L. . . . 78

S

Sachon, Michael . . . 107
Saz, Joan Rogalska . . . 305, 310, 364
Schaublin, Bradley K. . . . 372
Schelb, Ernst . . . 328
Scherer, Charlotte . . . 83
Scherer, Donald . . . 156
Schneebeck, Charles A. . . . 163
Schooley, Ralph E. . . . 306
Schwartz, Dan . . . 315
Schweitzer, Debra E. . . . 366
Scigliano, John A. . . . 268
Shafto, Sylvia . . . 88
Shatzer, Linda S. . . . 307
Sherman, Mark . . . 75
Shirk, Henrietta N. . . . 393
Shoemaker, Scott D. . . . 316
Signer, Barbara . . . 352

PROCEEDINGS INDEX

	Page
Sorisio, Linda B.	371
Spann, Melvin L.	14
Spannaus, Timothy W.	308
Spaulding, Aaron L.	317
Sprunger, Carol A. W.	210
Stanley, Charles J.	309
Starkweather, John A.	226
Stevens, David F.	305, 310
Stevens, Evelyn V.	117
Stevenson, Lynne	353
Suchinsky, Adele	37
Sutton, Rosemary E.	271

T

Thompson, Cecelia	127
Thompson, Eleanor G.	324
Thompson, Merton E.	170
Trainor, Mary Stoddard	354
Tremont, Paul J.	210
Trollip, Stanley	41

V

Vaughn, Gary	318
Vispoel, Walter	221
Vukelich, John	394

W	Page
Walters, B. Wayne	218
Waugh, Michael L.	319
West, Lois	311
White, Alan D.	45, 268
White, Arthur L.	276
Whiteman, Frederick C.	388
Wileman, Ralph	355
Williams, Tom	320
Wiltse, Clo	129
Woo, Mason	50
Wright, Pamela A.	321

Y

Yoder, Marianne E.	375

Z

Zrihen, Pamela J.	96

Printed by: ***OMNIPRESS*** ... ***"The Proceedings Printer"***
Madison, Wisconsin